THOMSON

TO7/70 GUIDE

Le guide du TO7 - 70

Le chapitre Comment fonctionne votre TO7-70
a été rédigé par André Deledicq et Serge Pouts-Lajus.

Illustrations : Raoul Raba
Maquette : Alain Dufourcq

Ce volume porte la référence
ISBN : 2-7124-0593-5

Toute reproduction, même partielle, de cet ouvrage est interdite. Une copie ou reproduction par quelque procédé que ce soit, photographie, photocopie, microfilm, bande magnétique, disque ou autre, constitue, une contrefaçon passible des peines prévues par la loi du 11 mars 1957 sur la protection des droits d'auteur.

© *CEDIC 1985*
CEDIC, 6-10 boulevard Jourdan, 75014 - Paris
Tél. : 565-06-06

SOMMAIRE

1 - Faites connaissance avec votre micro-ordinateur

Description extérieure 8
Installation 10
Mise en service 14
Pour utiliser le crayon optique 17
Pour utiliser une cartouche 20
Pour utiliser une cassette 21
Le clavier 22
La mémoire 27
Les extensions 29
Caractéristiques principales 32
Anomalies de fonctionnement et contrôles 34
Schéma synoptique de raccordement 35

2 - Comment fonctionnne votre TO7-70

Votre ordinateur 37
Principe de fonctionnement 38
Le transistor et la logique 40
Le codage binaire des informations 42
Le microprocesseur et les mémoires 46
La cartouche 51
Les périphériques 52
La mémoire d'écran 56
Que se passe-t-il sous le clavier du TO7-70 59
Micro dictionnaire 63

3 - Le moniteur du TO7-70

1.
FAITES CONNAISSANCE AVEC VOTRE MICRO-ORDINATEUR

Votre micro-ordinateur TO7-7Ø est spécialement conçu pour l'usage individuel ou familial et vous permettra :

— d'exécuter des programmes dans de multiples domaines :
- éducation assistée,
- gestion familiale,
- création de musique, d'images et de sons,
- jeux, etc. ;

— de programmer, selon la cartouche introduite dans l'ordinateur, en langage BASIC, LOGO, ASSEMBLEUR, FORTH, etc. ;

— de réaliser des dessins ou graphiques utilisant 16 couleurs différentes ;

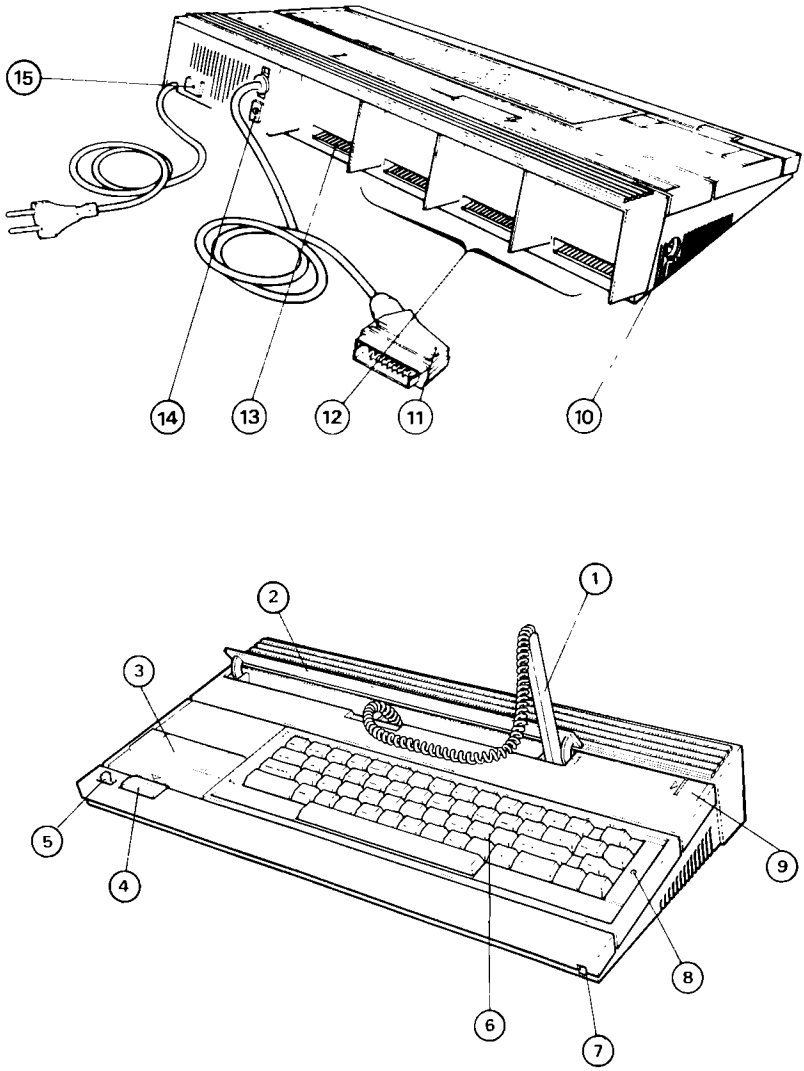
— de faire de la composition musicale sur 5 octaves ;

— de créer des dessins à l'aide du crayon optique intégré ;

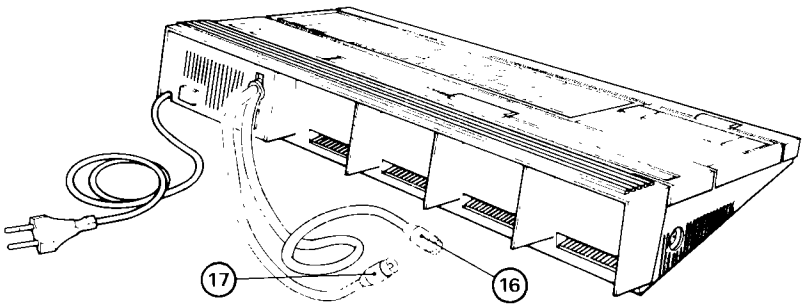
— d'incruster, à l'aide de l'interface d'incrustation et d'un téléviseur avec prise péritélévision, des images créées avec votre TO7-7Ø dans une image vidéo externe provenant :
- d'une émission de télévision
- d'un magnétoscope avec sortie vidéo
- d'une caméra vidéo.

DESCRIPTION EXTÉRIEURE

Pour vous permettre de vous familiariser avec votre micro-ordinateur, voici une description extérieure de l'appareil.



1 - Unité centrale CU70-PER



2 - Unité centrale CU70-PBG

1. Crayon optique.
2. Volet de la trappe du crayon optique.
3. Volet de la trappe destinée à recevoir les cartouches de programmes MEMO 7.
4. Touche d'ouverture du volet (3).
5. Commande de verrouillage de la touche d'ouverture (4).
6. Clavier permettant la frappe des caractères minuscules ou majuscules, des caractères spéciaux et des chiffres.
7. Voyant signalant la mise en service.
8. Voyant signalant que le clavier est en mode minuscule.
9. Touche permettant en cours d'utilisation de réinitialiser l'unité centrale (afficher le menu de départ).
10. Prise DIN pour le raccordement d'un lecteur-enregistreur de programmes.
11. Prise péritélévision permettant le raccordement au récepteur de télévision directement ou par l'intermédiaire d'un codeur-modulateur.
12. Connecteurs permettant le raccordement des modules d'extension.
13. Connecteur spécifique pour le raccordement de l'extension mémoire.
14. Prise 12 V pour l'alimentation d'un codeur-modulateur éventuel.
15. Touche de mise en service et d'arrêt de l'unité centrale.
16. Fiche DIN audio-vidéo : permet le raccordement au récepteur de télévision directement ou par l'intermédiaire d'un codeur-modulateur.
17. Fiche du câble d'antenne.

INSTALLATION

L'installation de votre micro-ordinateur doit obéir à quelques règles élémentaires :
Certains composants situés à l'intérieur de l'ordinateur chauffent et nécessitent d'être ventilés ; en conséquence, évitez de placer votre appareil près d'une source de chaleur et d'obstruer les ouvertures d'aération.

Raccordement avec votre téléviseur

Le raccordement entre votre micro-ordinateur et votre téléviseur ou moniteur s'effectue :
— si votre téléviseur ou moniteur est équipé d'une prise péritélévision :
à l'aide du câble péritélévision de votre TO7-7Ø à raccorder sur votre téléviseur (fig. 3 et 4).

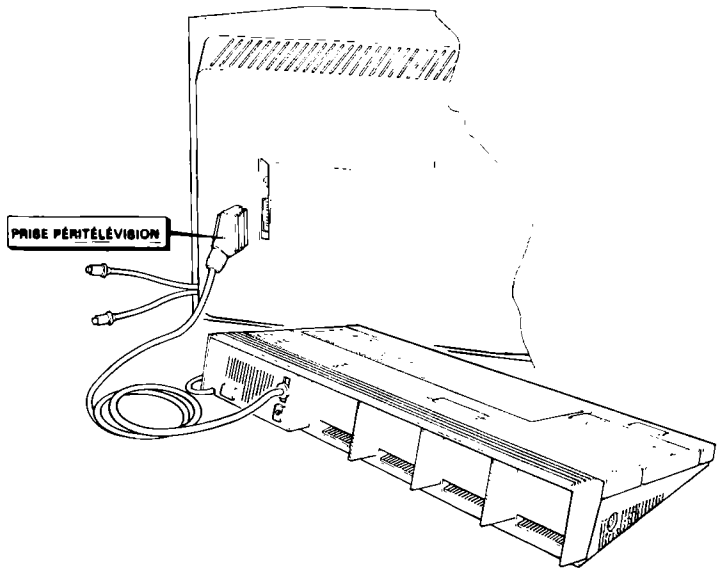
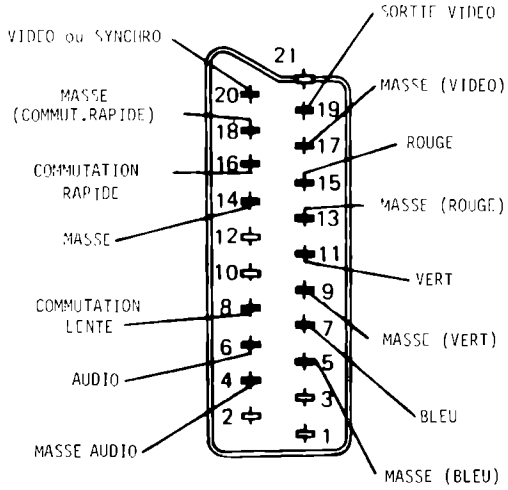


Fig. 3



BROCHAGE DE LA PRISE PÉRITÉLÉVISION
(côté utilisation) **Fig. 4**

– Si votre téléviseur ou moniteur n'est pas équipé d'une prise péritélévision et fonctionne au standard SECAM (France) :
par l'intermédiaire d'un codeur-modulateur disponible chez votre distributeur (fig. 5).

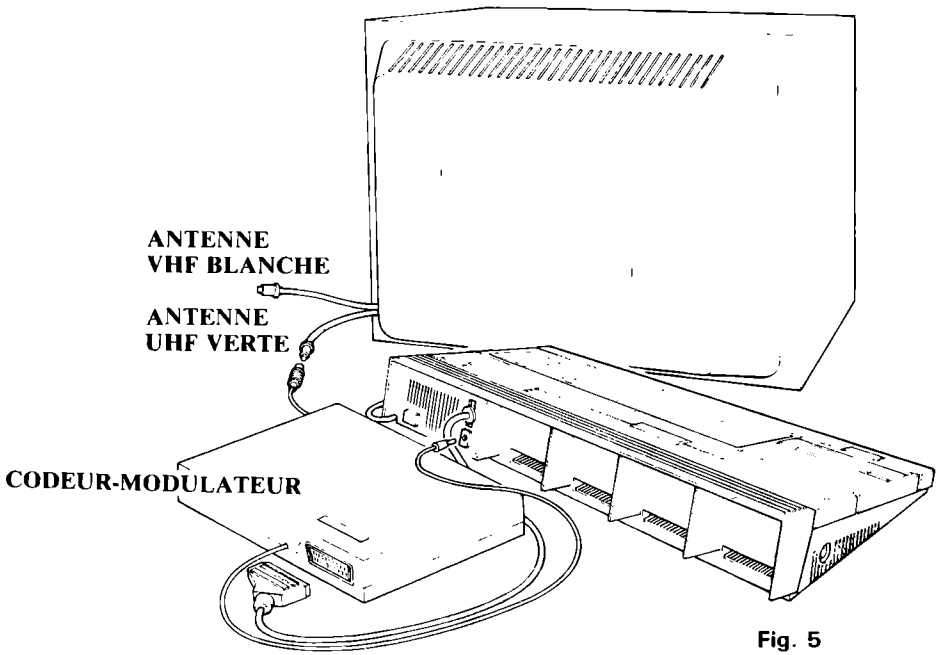


Fig. 5

— Pour les pays au standard P.A.L.

Si votre téléviseur ou moniteur est équipé d'une prise audio-vidéo DIN : à l'aide d'un câble disponible chez votre revendeur à raccorder d'un côté à la prise DIN du cordon de votre TO7-7Ø PBG et de l'autre à la prise DIN audio-vidéo de votre téléviseur (fig. 6 et 7).

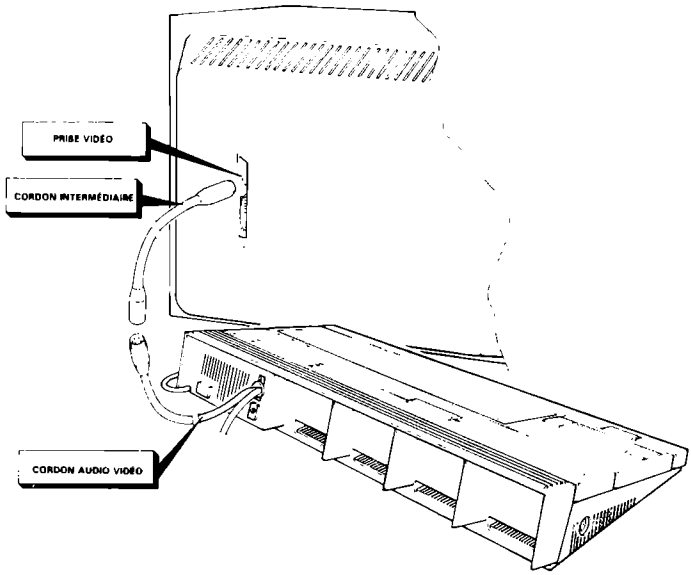
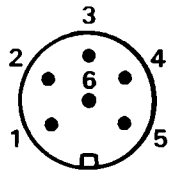


Fig. 6



BROCHAGE DE LA FICHE DIN FEMELLE AUDIO-VIDÉO
(côté utilisation)

1.5 + 12V
2 SORTIE VIDÉO
3 MASSE
4 SORTIE SON

Fig. 7

— Si votre téléviseur ou moniteur n'est pas équipé d'une prise péritélévision ou audio-vidéo :
à l'aide du câble d'antenne de votre TO7-7Ø PBG à raccorder d'un côté à la prise CINCH de votre TO7-7Ø et de l'autre sur l'entrée antenne UHF de votre téléviseur PAL (fig. 8).

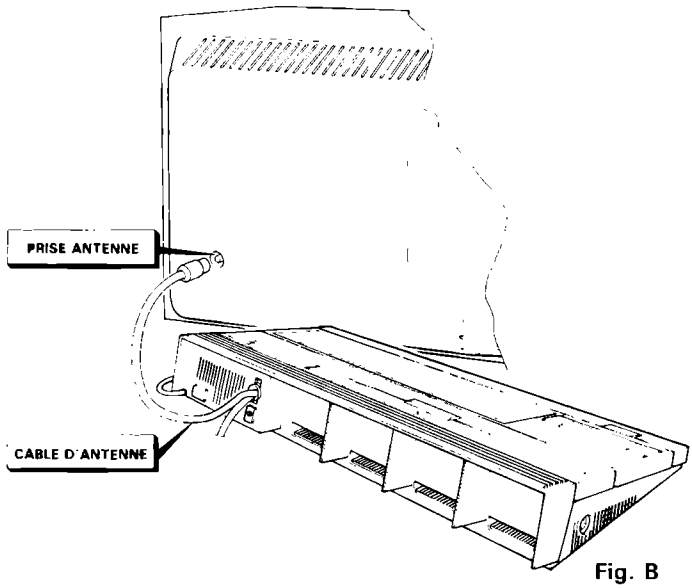


Fig. B

Raccordement au secteur

Branchez le cordon d'alimentation de votre micro-ordinateur sur une prise secteur 220 V ou 240 V 50 Hz.

Raccordement du lecteur-enregistreur de programmes

Branchez le cordon muni d'une prise DIN sur la prise n° 10 (fig. 1) de votre micro-ordinateur et le cordon d'alimentation sur une prise secteur 220 V ou 240 V 50 Hz.

Raccordement du contrôleur-lecteur-enregistreur de disquettes

Branchez le boîtier d'interface de votre contrôleur sur un emplacement d'extension à l'arrière de votre micro-ordinateur et reliez le cordon d'alimentation à une prise secteur 220 V ou 240 V 50 Hz.

MISE EN SERVICE

La mise sous tension de votre installation doit être réalisée dans l'ordre suivant :

a) le récepteur de télévision ou le moniteur,

b) les différentes extensions (par exemple les unités de disquettes, l'imprimante, etc.),

c) l'ordinateur.

— Dans le cas de l'utilisation d'un téléviseur ou d'un moniteur ne possédant ni prise péritélévision ni prise Audio-Vidéo :
vous devez sélectionner, s'il existe, le bouton de programme spécialisé Audio-Vidéo (consultez le guide d'utilisation de votre téléviseur) et procéder au réglage de la façon suivante :
Sélectionnez le canal 35 du téléviseur et continuez lentement le réglage vers le canal 36. Vous allez obtenir successivement une première image couleur de qualité médiocre avec du bruit dans le haut-parleur, puis une image en noir et blanc et enfin une image très stable en couleur qui indique le bon réglage.

— Dans le cas de l'utilisation d'un téléviseur ou d'un moniteur équipé, soit d'une prise Audio-Vidéo, soit d'une prise péritélévision :
l'image doit apparaître dès la mise en service de votre TO7-7Ø éventuellement après avoir sélectionné le bouton de programme Audio-Vidéo s'il existe. (Notez que le branchement de la prise péritélévision interrompt sur la plupart des téléviseurs les réglages de couleur et parfois du contraste) (fig. 9).

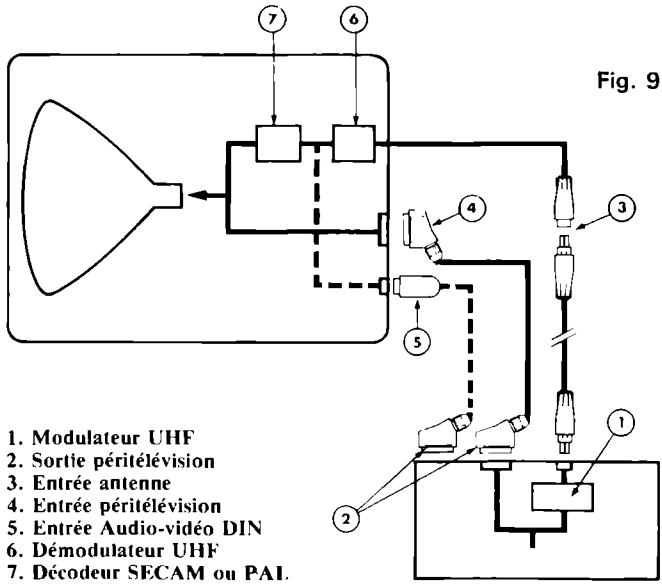
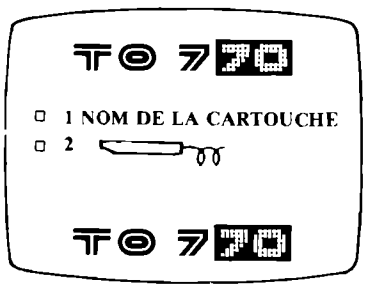


Fig. 9

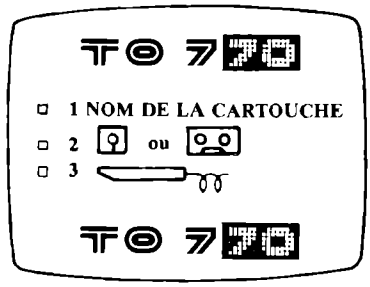
1. Modulateur UHF
2. Sortie péritélévision
3. Entrée antenne
4. Entrée péritélévision
5. Entrée Audio-vidéo DIN
6. Démodulateur UHF
7. Décodeur SECAM ou PAL

La qualité de l'image obtenue est, comme vous pouvez le remarquer, très sensiblement supérieure à celle obtenue par l'utilisation de l'entrée antenne, dans la mesure où ces prises spéciales évitent de retransformer l'image issue de votre TO7-7Ø en un signal identique à celui fourni par votre antenne.

Sur l'écran apparaît le menu du TO7-7Ø :



Sans périphérique



Avec périphérique

Celui-ci contient une liste des possibilités qui vous sont offertes :

1. Le nom de la cartouche qui est engagée (voir page 20, mise en place d'une cartouche de programmes).
2. Un pictogramme qui représente un périphérique branché :
   - soit : lecteur de cassette,
   - soit : contrôleur-lecteur de disquettes.

Lorsque les deux appareils sont branchés, le contrôleur-lecteur de disquettes est reconnu en priorité.

3. Le pictogramme du crayon optique représente le crayon optique intégré dans votre TO7-7Ø.

Démarrage du TO7-70

Selon le choix que vous réalisez sur le menu, vous demanderez l'exécution d'une action liée au symbole retenu.
Le choix peut être réalisé :

- soit au clavier en frappant le numéro choisi,
- soit avec le crayon optique en pointant le carré en face du numéro choisi.

Conséquence du choix

1. Vous demanderez l'exécution du logiciel de la cartouche enfichée dans la trappe. Pour plus de détails, consultez la documentation de la cartouche de programmes.
2. Vous désirez exécuter un logiciel stocké sur cassette ou disquette :
   - sur cassette : vous allez exécuter le premier programme qui va être lu,
   - sur disquette : vous allez exécuter un programme qui a pour nom "AUTO. BAT".

ATTENTION : L'exécution du programme n'est réalisée qu'à la condition qu'il soit écrit dans le langage qui correspond à la cartouche.

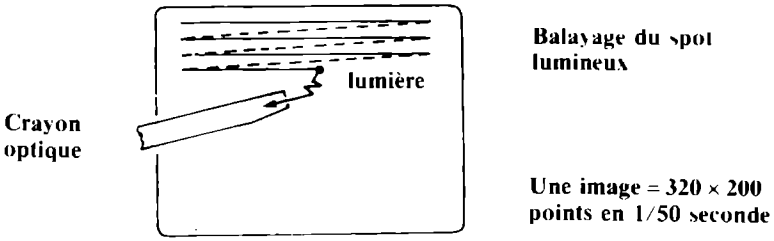
3. Vous demandez le réglage du crayon optique.

POUR UTILISER LE CRAYON OPTIQUE

Le crayon optique et son câble de liaison sont rangés dans une trappe située sur le dessus de l'ordinateur. En position de travail, le couvercle de la trappe reste ouvert et un logement permet de placer le crayon en position verticale.
Le crayon optique accroît les possibilités de votre ordinateur ; il vous permet de dessiner ou de sélectionner des informations sur l'écran de votre téléviseur et d'établir un véritable dialogue interactif avec l'ordinateur.
Le crayon optique réalise deux fonctions indépendantes :
1. une localisation,
2. une validation.

Localisation

L'image apparaissant à l'écran de votre téléviseur sous commande de l'ordinateur est constituée par le balayage d'un spot lumineux sur 312 lignes, 50 fois par seconde.
La localisation consiste à repérer la position relative du crayon sur l'écran, c'est-à-dire à détecter le passage du spot lumineux lors du balayage.
Ceci est réalisé grâce à un capteur sensible à la luminosité du téléviseur (element photo-sensible).



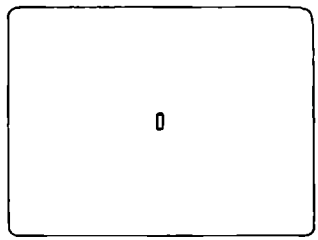
Un mauvais réglage de la luminosité du téléviseur peut entraîner une saisie difficile des informations. D'autre part, celles-ci ne sont pas prises en compte lorsque la zone pointée est noire ou rouge.

Validation

La validation est assurée par un interrupteur logé dans la pointe du crayon. Le contact se ferme lorsque l'on appuie le crayon sur une surface plane telle que l'écran du téléviseur.
Sous contrôle d'un programme, il est possible de lire les coordonnées du point visé sur l'écran et, d'autre part, de savoir si le contact est enclenché ou non.
Le crayon optique peut donc être très utile dans des applications interactives telles que : conception, dessin, jeux et enseignement assisté.

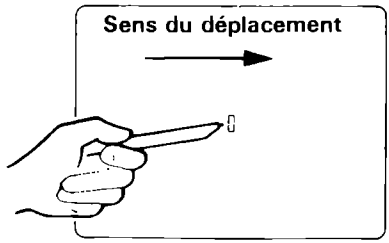
Réglage

Une mire de réglage permet à l'ordinateur d'adapter le crayon optique à tous les téléviseurs. En particulier, elle assure une synchronisation correcte entre l'information de détection du spot lumineux fournie par le crayon otpique et le signal de balayage du téléviseur.
La méthode pour faire apparaître la mire de réglage est indiquée sur le menu proposé à l'écran soit à la mise sous tension de l'ordinateur, soit à la suite d'une initialisation.



Mire de réglage du crayon optique

Pour effectuer le réglage :
— passer le crayon optique devant le segment blanc en exerçant une légère pression; un "bip" se fait entendre et le menu apparaît de nouveau : le réglage est effectué.



Remarques

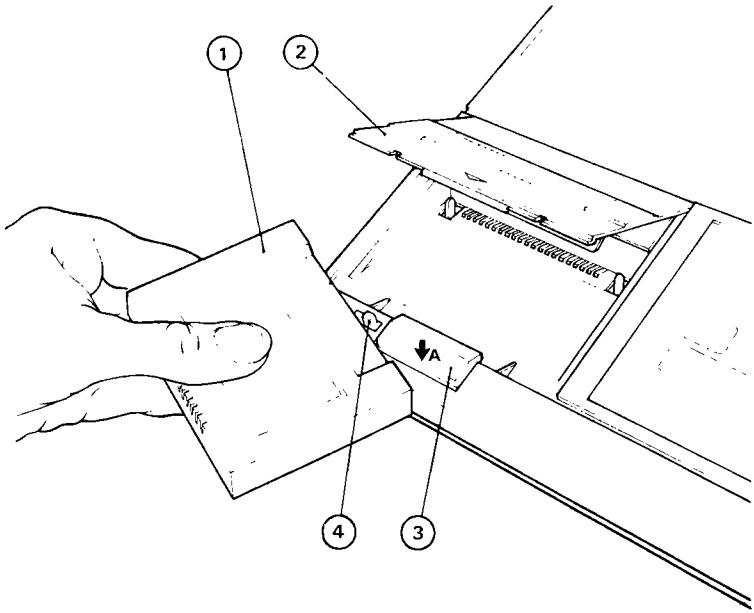
• Il est nécessaire de refaire le réglage du crayon optique lorsque l'alimentation de l'ordinateur a été coupée.
• Il est recommandé de tenir le crayon optique perpendiculairement à l'écran afin d'éviter une erreur de lecture.

Rangement du crayon optique

Une attention particulière doit être portée au rangement du crayon optique dans l'emplacement qui lui est réservé.
Il faut prendre garde au fait que le bouton-poussoir doit rester libre.
Si le bouton-poussoir du crayon optique reste appuyé, l'initialisation du clavier se trouve bloquée.

POUR UTILISER UNE CARTOUCHE

La mise en place ou le retrait d'une cartouche de programme doit toujours s'effectuer le TO7-7Ø éteint et l'étiquette sur le dessus.
De par sa conception, votre micro-ordinateur ne peut fonctionner que s'il est pourvu d'une cartouche de programmes. C'est en effet dans ces cartouches enfichables que résident les différents programmes de jeux, de gestion, d'éducation assistée ainsi que les langages qui vous permettront d'écrire vous-même vos programmes.
Vous devez procéder de la façon suivante :
— assurez-vous que votre micro-ordinateur n'est plus sous tension,
— déverrouillez la commande (4) en la poussant vers la gauche,
— appuyez sur la touche (3) dans le sens de la flèche (A), la trappe (2) se soulève,
— présentez la cartouche avec l'étiquette sur le dessus comme indiqué sur la figure,
refermez la trappe (2), et appuyez pour la verrouiller,
poussez la commande (4) vers la droite afin de verrouiller la touche (3),
pour retirer la cartouche, procédez de la manière inverse.



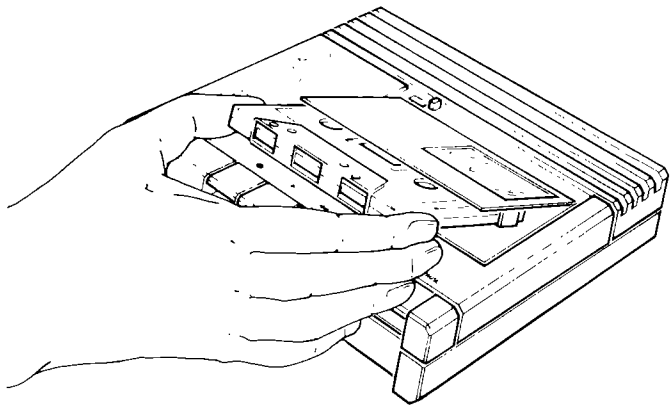
Légendes
1. cartouche
2. trappe
3. touche d'ouverture
4. verrou de trappe

POUR UTILISER UNE CASSETTE

L'utilisation du lecteur-enregistreur de programmes nécessite la présence d'une cartouche dans le TO7-7Ø.

— Pour transférer un programme sur cassette vers votre TO7-7Ø :
  - placez la cassette dans le LEP;
  - appuyez sur la touche lecture;
  - frappez 2 ou pointez le carré blanc devant le 2 avec le crayon optique;
  - attendez que le programme sur cassette entre dans la mémoire du TO7-7Ø et suivez les instructions données avec la cassette;
  - vous noterez que le LEP est un magnétophone spécifique qui vous permet, sur certaines cassettes, d'entendre de la musique et des commentaires pendant le transfert du programme enregistré vers la mémoire du TO7-7Ø.

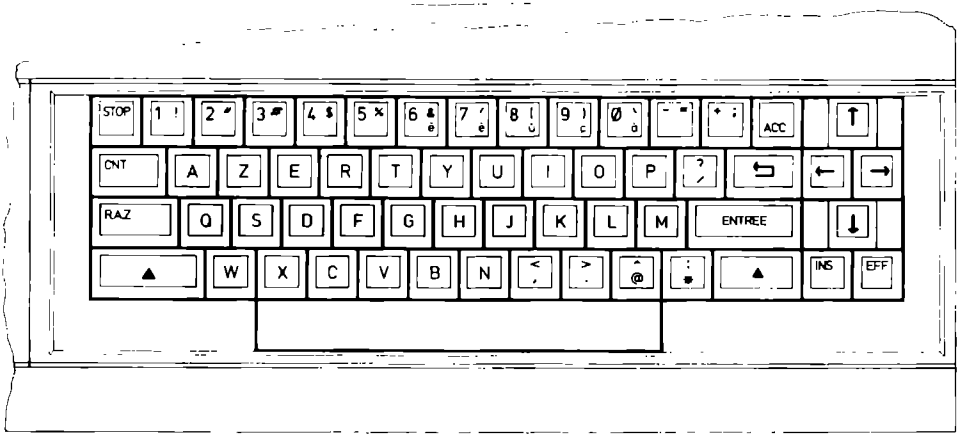
— Pour transférer un programme de votre TO7-7Ø sur une cassette :
  - suivez les instructions de votre cartouche de langage.



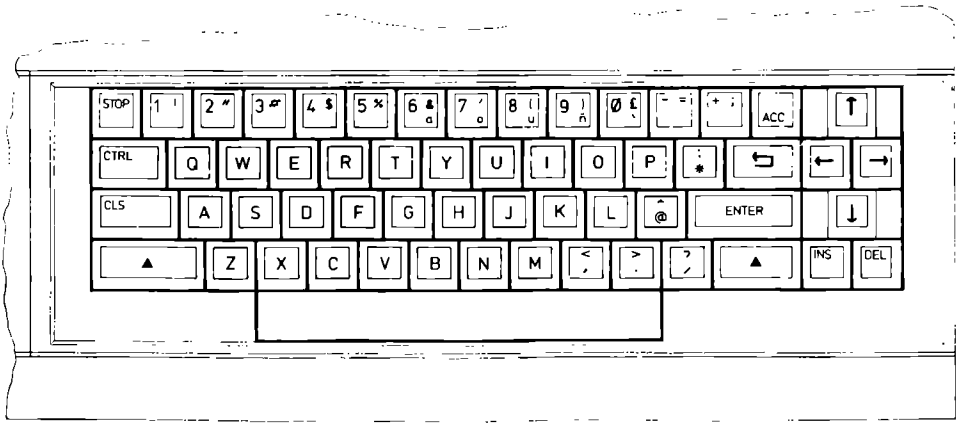
LA CLAVIER

Le clavier constitue le principal moyen de communication avec l'ordinateur. Il permet l'introduction d'informations et de programmes dans l'unité centrale et l'établissement d'un dialogue avec cette dernière.
Le clavier du TO7-7Ø est du type mécanique.
Il comporte :
— 14 touches de fonction ;
— 43 touches pour les symboles et caractères alphanumériques ;
— une barre d'espacement.
Chaque fois qu'une touche est sollicitée, un "bip" sonore est transmis à la voie son du téléviseur.
L'action prolongée sur une des touches permet, après une temporisation de une seconde, la répétition de la touche à la cadence de 10 fois par seconde.

Clavier AZERTY



Clavier QWERTY



Les touches de fonction

CNT ou CTRL : Cette touche "CONTRÔLE" doit être utilisée simultanément avec une autre touche dont elle modifie le sens (elle est utilisée particulièrement par l'éditeur de langage BASIC).

: MAJUSCULE
Cette touche doit être utilisée simultanément avec une autre touche dont elle modifie le sens.

et barre d'espacement : Permet de passer de façon permanente du mode écriture majuscule au mode écriture minuscule ou inversement.

et touche double : Introduit les symboles marqués en vert (quel que soit le mode d'écriture, minuscule ou majuscule).

et touche alphabétique : Introduit une lettre majuscule alors que l'ordinateur est en mode écriture minuscule.

NOTA : Les deux touches placées de chaque côté du clavier ont la même fonction.

ENTREE ou ENTER : L'action sur cette touche permet de valider une commande ou une ligne de programme, puis de redonner le contrôle à l'utilisateur.

Barre d'espacement : Elle envoie le caractère "ESPACE" (parfois appelé "blanc"). Elle est utilisée pour séparer des mots ou des expressions.

ACC : ACCENT
L'action sur cette touche constitue le premier temps d'une séquence permettant l'introduction de caractères ou symboles particuliers et notamment les minuscules accentuées.

A - Frappe en 2 temps :

— Appuyez sur la touche [ACC].

— Appuyez sur la touche correspondant au caractère à introduire.

• Frappez en 2 temps. Cette liste est limitée à :

Clavier AZERTY

CARACTERE A INTRODUIRE	1er TEMPS FRAPPE DE LA TOUCHE	2e TEMPS FRAPPE DES TOUCHES
é	ACC	6
è		7
ù		8
ç		9
à		0

Clavier QWERTY

CARACTERE A INTRODUIRE	1er TEMPS FRAPPE DE LA TOUCHE	2e TEMPS FRAPPE DE LA TOUCHE
ä	ACC	6
ö		7
ü		8
ñ		9

B - Frappe en 3 temps

— Passez en mode écriture minuscule en appuyant simultanément sur les touches **barre d'espacement** et [●]
— Appuyez sur la touche [ACC].
— Appuyez sur la touche correspondant à l'accent à introduire.
— Appuyez sur la touche correspondant à la lettre à introduire.

• Frappez en 3 temps. Voici à titre d'exemple :

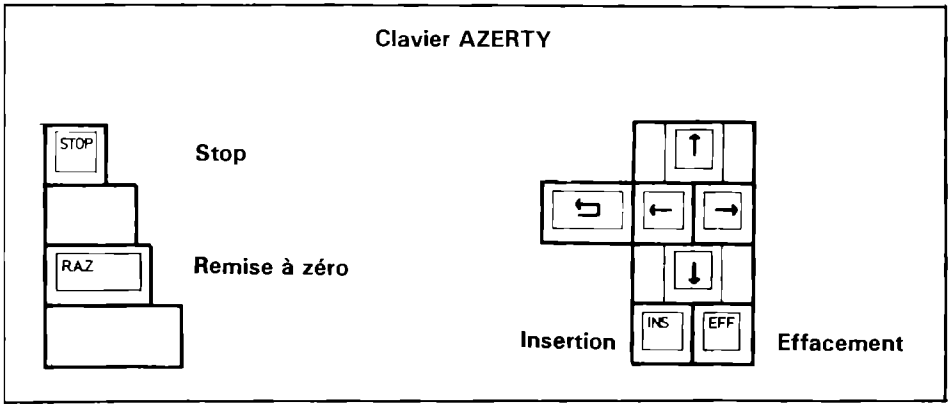
Clavier AZERTY

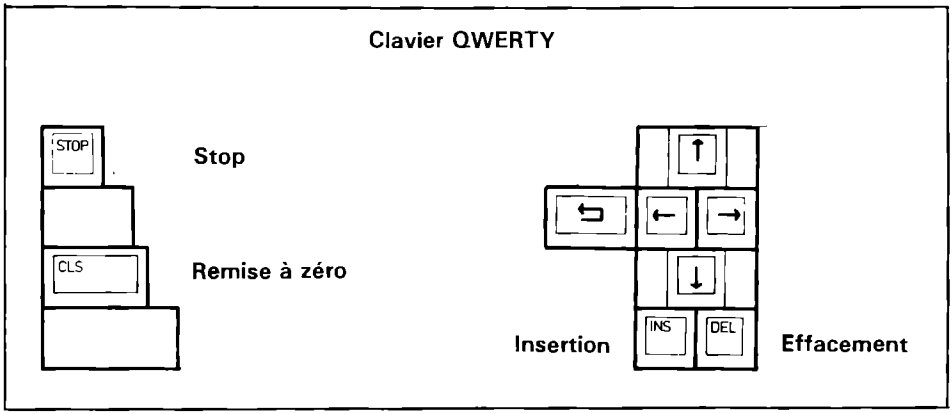
CARACTERE A INTRODUIRE	MODE D'ECRITURE	1er TEMPS FRAPPE DE LA TOUCHE	2e TEMPS FRAPPE SIMULTANEE DES TOUCHES	3e TEMPS FRAPPE DE LA TOUCHE
é			[●] [7]	[E]
è			[●] [0]	[E]
ï	minuscule	[ACC]	[●] [2]	[I]
â			[●] [^ @]	[A]

Clavier QWERTY

CARACTERE A INTRODUIRE	MODE D'ECRITURE	1er TEMPS FRAPPE DE LA TOUCHE	2e TEMPS FRAPPE SIMULTANEE DES TOUCHES	3e TEMPS FRAPPE DE LA TOUCHE
ä			[●] [2]	[A]
ö			[●] [2]	[O]
ü	minuscule	[ACC]	[●] [2]	[U]
ñ			[●] [2]	[N]
caractère à accentuer			l'accent	lettre minuscule

Touches dont la fonction dépend du programme (ou de la cartouche). La notice jointe à chaque programme explique la fonction particulière qu'il peut affecter à certaines touches du clavier.

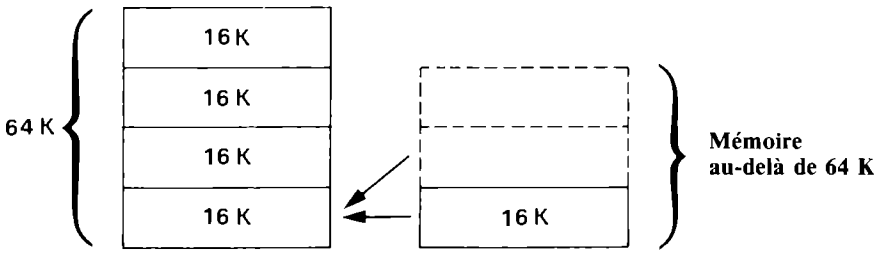




LA MÉMOIRE

Principe d'organisation

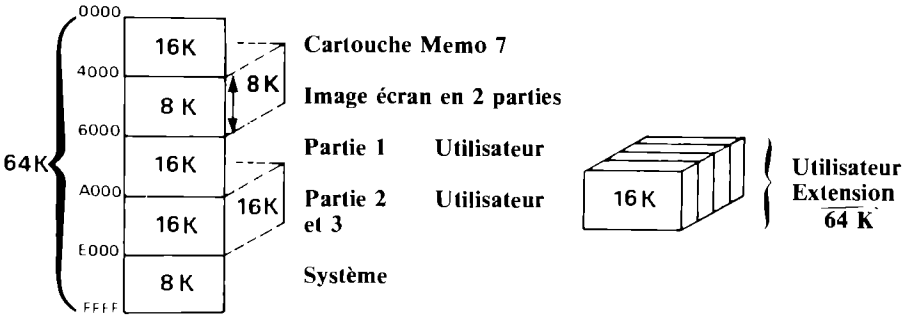
De par son principe d'adressage de la mémoire, un ordinateur tel que le TO7 (microprocesseur 8 bits) ne peut adresser qu'un volume mémoire de 64 K. Pour le TO7-7Ø, une solution technique permet d'adresser une mémoire beaucoup plus importante. Le principe consiste à considérer qu'au-delà de 64 K, une partie de la mémoire se superpose à la mémoire de base. Pour l'ordinateur, la mémoire gérée est toujours 64 K mais physiquement, il ne s'agit pas toujours de la même partie.
Ce transfert fictif se réalise par bloc de mémoire (un bloc étant de 16 K).



Fonctionnement

• Pour l'utilisation de la mémoire au-delà de 64 K, il suffit donc de préciser quel bloc de mémoire viendra se placer fictivement à la place d'un même bloc de mémoire de base.
• La partie mémoire utilisateur du TO7-7Ø contient 48 K en trois parties :
Partie 1 → 16 K contenus dans les 64 K de base.
Partie 2 → 16 K contenus dans les 64 K de base. Cette partie est fictivement recouverte par les autres.
Partie 3 → Il s'agit d'un bloc mémoire utilisable uniquement en recouvrement.

Schéma mémoire du TO7-70



Utilisation

La partie de mémoire utilisateur (partie 3) comprise entre 32 et 48 K est utilisable en recouvrement à condition de commander ce transfert par programmation. L'utilisation de cette mémoire est donc liée au programme contenu dans la cartouche.

Extension mémoire

L'extension mémoire (présentée parmi les extensions) est composée de 4 blocs de mémoire, soit 64 K. Pour son utilisation, chaque bloc est mis en recouvrement de la partie 2 de la mémoire utilisateur.

LES EXTENSIONS

Comme pour la mise en place d'une cartouche de programmes, IL EST INDISPENSABLE DE COUPER L'ALIMENTATION DE L'ORDINATEUR AVANT DE CONNECTER OU DE RETIRER UNE EXTENSION.
Chaque programme vendu par notre distributeur comporte sur son emballage la liste des extensions nécessaires ou optionnelles pour tirer pleinement parti des possibilités apportées par ce programme.
Un certain nombre d'extensions disponibles chez notre distributeur permettront de rendre votre ordinateur encore plus performant. La liste des extensions ci-dessous n'est pas limitative.

Contrôleur de communication

Celui-ci assure une liaison série, RS232, et une liaison parallèle.
Tous les terminaux, imprimantes, tablettes graphiques, etc., peuvent ainsi être raccordés à l'ordinateur à travers cette extension.

Extension mémoire

Pour accroître la capacité mémoire de votre ordinateur, un module enfichable sur le connecteur 26 broches situé à l'arrière de l'unité centrale permet de porter la capacité de la mémoire utilisateur de 48000 caractères à 112000 caractères (112 K mémoire vive).

REMARQUE :
Cette extension nécessite des conditions particulières d'utilisation. Le principe d'adressage de l'extension mémoire est identique à l'adressage d'une partie de la mémoire utilisateur. La description en a été faite au chapitre MÉMOIRE.
L'utilisation de cette extension est liée aux caractéristiques de la cartouche de programmes MÉMO 7.

Imprimante

Une imprimante vous permettra de conserver sur papier les programmes que vous aurez écrits ou le résultat de leur exécution.
— imprimantes à impact 80 colonnes ;
NOTA : Les instructions relatives à la commande d'une imprimante propres à chaque cartouche de programmes sont exposées dans la notice de celle-ci.

Lecteur-enregistreur de programmes

De par sa facilité d'emploi et sa rapidité d'exécution, le lecteur-enregistreur de programmes vous sera d'une grande utilité pour sauvegarder vos programmes, pour charger dans l'ordinateur ceux que vous aurez achetés* ou pour stocker des données.
Une deuxième piste est réservée à la restitution de sons.
NOTA : Seuls les lecteurs-enregistreurs de programmes portant le sigle TO7 peuvent être utilisés avec l'ordinateur TO7-7Ø.

Contrôleur et lecteur de disquettes

L'ensemble contrôleur et lecteur de disquettes permet l'enregistrement de programmes, comme le lecteur-enregistreur de cassettes, mais de manière beaucoup plus rapide et plus souple. De même, il est possible de stocker et de gérer des données (liste de nombres, de mots...) en quantité beaucoup plus grande que celles de la mémoire de l'unité centrale et de manière plus souple que sur le lecteur de cassettes.
Il vous permet également d'exploiter des logiciels pré-enregistrés sur disquettes, en particulier la disquette DOS BASIC du TO7-7Ø rend plus performante la cartouche BASIC par l'adjonction de nouvelles instructions et fonctions (de même pour FORTH, LOGO...).

Extension musique et jeux

Elle est constituée d'un module et de deux manettes (référence CM 90-112).
Un générateur de fréquence permet, sous contrôle d'une cartouche de programmes spécifique, de synthétiser des sons qui seront reproduits par l'intermédiaire du téléviseur sur une, deux, trois ou quatre voix.
Suivant l'inclinaison de la commande des manettes de jeux, différents ordres sont générés et transmis vers l'ordinateur qui les interprète et les exécute. Un bouton-poussoir permet, suivant la nature du programme de jeux, le déclenchement d'un tir, l'envoi d'une balle, etc.

Extension télématique

Cette extension comprend :
— un MODEM (modulateur-démodulateur) rendant compatible des signaux numériques avec des informations de fréquence transmissibles par voie téléphonique.
— un câble de raccordement direct à la ligne téléphonique.
Les caractéristiques sont conformes aux normes appliquées pour le système VIDÉOTEX français appelé TÉLÉTEL.

*Seuls les programmes pré-enregistrés portant la mention TO7 peuvent être lus par votre ordinateur.

Cette extension permet la consultation des banques de données (annuaire électronique, TÉLÉTEL, etc.) par l'intermédiaire de votre ordinateur. Le raccordement à la ligne PTT s'effectue directement entre la prise téléphonique murale et le téléphone. Une cartouche enfichable fournit le logiciel nécessaire à l'interrogation des banques de données.

Extension incrustation

Ce module permet de mixer l'image TO7-7Ø avec celle issue de toute source vidéo externe : émission TV, caméra vidéo, magnétoscope, vidéoscope, etc.

CARACTÉRISTIQUES PRINCIPALES

Présentation

COFFRET	: matière moulée.
CLAVIER	: type mécanique.
DIMENSIONS	: L. 455 - H. 75 - P. 260 mm.
MASSE	: — CU70-PER : 2,6 kg — CU70-PBG : 2,6 kg.

Particularités électriques

ALIMENTATION	: secteur 220 V - 50 Hz.
CONSOMMATION	: 20 W (sans extension) 35 W (avec toutes les extensions).
FUSIBLE	: T 800 mA.

Une étiquette fixée sous l'appareil reproduit les particularités électriques mentionnées ci-dessus.

Particularités techniques

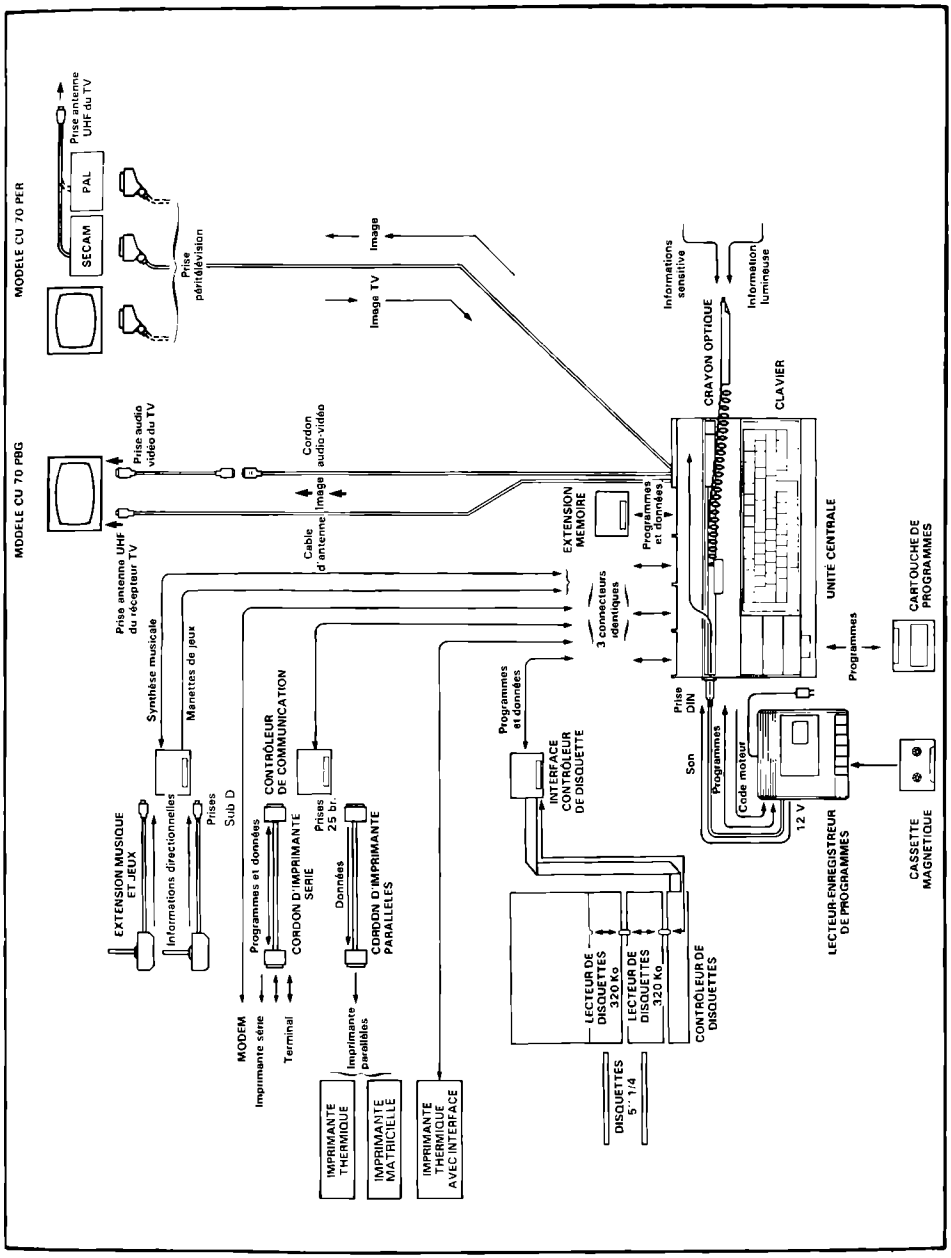
MICROPROCESSEUR	: 6809E - 1 MHz
MÉMOIRE 70 K	: Utilisateur : 48 K (mémoire vive) extensible à 112 K (mémoire vive) Écran graphique : 16 K (mémoire vive) Moniteur : 6 K (mémoire morte).
ALIMENTATION	: + 5 V continu 1,8 A + 12 V continu 300 mA − 5 V continu 100 mA.
ÉCRAN	: Sortie RVB + Son par prise péritélévision 320 × 200 points - 16 couleurs 25 lignes × 40 caractères Majuscules et minuscules accentuées.
CLAVIER	: 57 touches QWERTY avec accents ou 57 touches AZERTY avec accents.
MUSIQUE	: Synthèse musicale sur 5 octaves.
CRAYON OPTIQUE	: Résolution 320 × 200 points.

INTERFACE CASSETTE	: 900 bauds avec télécommande pour lecteur-enregistreur Voie sonore sur la deuxième piste.
PRISE ANTENNE (CU70-PBG)	: — Sortie UHF — Fréquence 591 MHz — Z = 75 Ω.
PRISE DIN (CU70-PBG)	: — Signal vidéo 1 V c/c 75 Ω — Signal vidéo 1 V - Z > 10 kΩ.
PRISE PÉRITÉLÉVISION (CU70-PER)	: Sorties RVB + Son.

ANOMALIES DE FONCTIONNEMENT ET CONTRÔLES

ANOMALIES	CONTRÔLES
Le voyant "MARCHE-ARRÊT" ne s'allume pas à la mise sous tension	— S'assurer que la tension d'alimentation secteur de l'unité centrale est comprise entre 198 V et 242 V. — Vérifier le cordon et la prise d'alimentation secteur. — Contrôler le fusible secteur.
Le "menu" n'apparaît pas à l'écran du téléviseur	— Contrôler les réglages du téléviseur. — Vérifier les connexions à la prise péritélévision du récepteur ou éventuellement à celles du codeur-modulateur.
Le nom de la cartouche de programmes n'apparaît pas sur l'écran	— La trappe du lecteur de cartouche est incorrectement verrouillée. — Effectuer une initialisation programme.
Barres noires ou mauvaise synchronisation de l'image. Blocage clavier. L'unité centrale ne réagit à aucune commande.	— Parasites secteur ou rayonnement magnétique proche. — S'assurer que le téléviseur ou les éventuels périphériques ont été mis sous tension avant l'unité centrale. — Effectuer une initialisation programme.
Pas de son.	— S'assurer de la présence du "BIP" sonore du clavier et vérifier la commande de volume du téléviseur. — Vérifier les connexions à la prise péritélévision du récepteur ou éventuellement à celles du codeur-modulateur.
Mauvais fonctionnement du lecteur-enregistreur de programmes après raccordement à l'unité centrale.	— Vérifier que le lecteur-enregistreur fonctionne sans être raccordé à l'ordinateur. —Vérifier son cordon de raccordement ou le branchement à la prise "LECTEUR-ENREGISTREUR DE PROGRAMMES". — Contrôler la nature du logiciel d'application inséré dans le lecteur de cartouches.
Le lecteur-enregistreur de programmes ne charge pas de programmes en mémoire centrale ou n'enregistre ni des données ni des programmes.	— Vérifier les commandes du lecteur-enregistreur de programmes. — Consulter les manuels d'utilisation et de logiciel qui expliquent la gestion de ce périphérique.
Manque de sensibilité du crayon optique.	— Vérifier le réglage de luminosité du téléviseur et effectuer le réglage du crayon optique. — S'assurer que la zone pointée n'est ni rouge ni noire.
Blocage du clavier.	— Vérifier que le crayon optique est correctement rangé dans son logement et que le bouton-poussoir n'est pas appuyé.

SCHÉMA SYNOPTIQUE DE RACCORDEMENT



2.
COMMENT FONCTIONNE VOTRE TO7-70

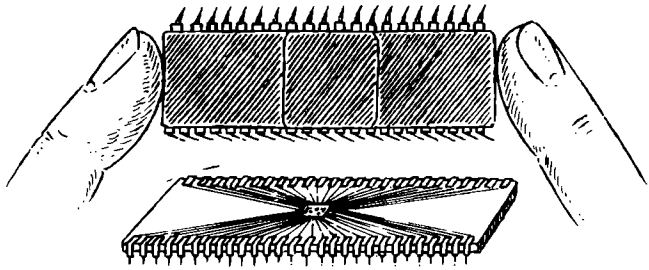
VOTRE ORDINATEUR

Le voici prêt à vous servir! Vous allez le brancher.
Le clavier va vous permettre de lui donner des ordres, l'écran vous prouvera que vous avez été compris et obéi.
Votre ordinateur est une machine.
D'abord une machine à écrire où la feuille blanche serait remplacée par un écran. Mais surtout, une machine à calculer ultra-rapide grâce à laquelle vous pouvez bien entendu faire des calculs mais aussi créer des images, des sons, des jeux. Entre le clavier et l'écran, c'est le domaine de l'électronique, du matériel informatique, de la technique.
Votre micro-ordinateur est un appareil électrique.
Sa vocation est de traiter des informations codées sous forme d'impulsions électriques.
Le montage matériel qui permet le traitement et la logique de son organisation sont d'une très grande complexité. Il n'est pas question ici d'en faire l'inventaire mais d'indiquer sommairement les grands principes qui font de l'informatique le plus puissant outil de traitement automatique de l'information.
On a coutume de séparer le problème en deux parties : le matériel et le logiciel, le "hard" et le "soft" pour les Anglo-saxons. C'est-à-dire la technique et la logique. D'une part, les composants électroniques, d'autre part les programmes qui mettent la machine au service des besoins humains.
Inutile de dire que matériel et logiciel ne sauraient se passer l'un de l'autre et que les progrès de l'un accompagnent les progrès de l'autre.

PRINCIPE DE FONCTIONNEMENT

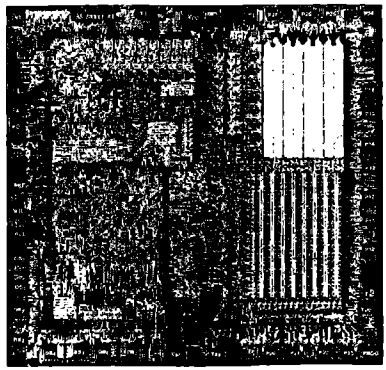
Le microprocesseur du TO7-70

Toute l'activité du micro-ordinateur est sous le contrôle d'un circuit intégré spécialisé, le microprocesseur. Les possibilités de la machine s'identifient pour beaucoup aux possibilités du microprocesseur.
Sur cette photo prise sous le clavier du TO7-7Ø, le microprocesseur se distingue par la taille; 40 "pattes" lui permettent de communiquer avec les autres circuits spécialisés.



Le Microprocesseur

L'apparition des microprocesseurs représente l'une des conquêtes marquantes de l'électronique moderne. L'avancée technique qu'ils réalisent porte sur la vitesse et l'intégration : un nombre considérable (jusqu'à un million) de composants rassemblés sur une surface très faible (25 mm^2) permet des vitesses de transmission électrique très élevées grâce à la réduction des distances.
Le microprocesseur est fabriqué à partir d'une plaque de silicium dans laquelle sont gravés des circuits et des transistors. Il est bien évident que la technique de fabrication utilisée fait appel à des innovations qui distinguent nettement les composants des microprocesseurs et ceux des radiotransistors; a fortiori des vénérables postes à lampes de nos aïeux.



La microphotographie du microprocesseur donne une idée précise de l'extraordinaire complexité du circuit qu'il contient.

Comment cette petite pastille peut-elle faire ce que l'on appelle "traiter des informations"?
Répondre à cette question c'est évidemment expliquer une grande partie du phénomène informatique, et c'est ce que vont tenter les quelques pages qui suivent!

LE TRANSISTOR ET LA LOGIQUE BINAIRE

Qu'est-ce qu'un transistor ?

Le microprocesseur est composé, nous venons de le voir, d'un nombre très important de petites unités que l'on appelle "transistors" qui agissent à la manière d'un interrupteur, ou plus exactement d'un robinet :

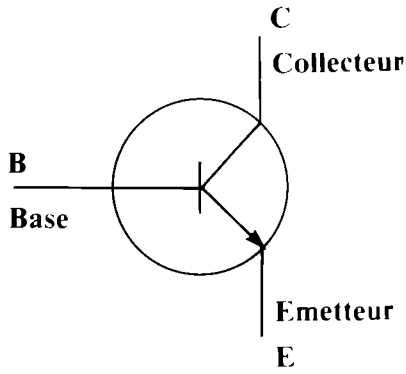
- si le robinet est ouvert, l'eau passe,
- si le robinet est fermé, l'eau ne passe pas.

Le transistor fonctionne de manière identique :

- la commande du robinet s'appelle la base,
- le tuyau d'arrivée d'eau s'appelle le collecteur,
- le tuyau de sortie d'eau s'appelle l'émetteur.

En considérant le courant électrique comme un flux d'électrons (charges élémentaires négatives), l'analogie avec le robinet peut être poussée jusqu'à son terme :

- la commande de l'interrupteur s'appelle la base,
- l'arrivée des électrons s'appelle l'émetteur,
- la sortie des électrons s'appelle le collecteur.



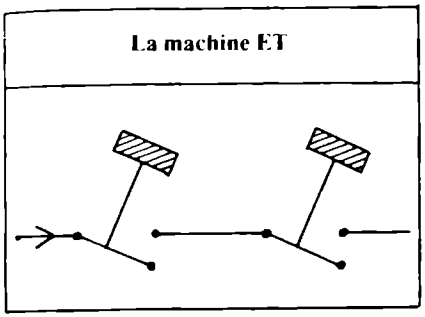
Un bit (c'est-à-dire un paquet d'électrons) se présente à la branche "émetteur" E. Si rien ne se passe dans la branche B, alors rien ne passe de E vers C (l'interrupteur est "ouvert").
Par contre, si des électrons sont "aspirés" par la branche B, alors l'interrupteur se ferme et le courant s'établit de l'émetteur E vers le collecteur C.

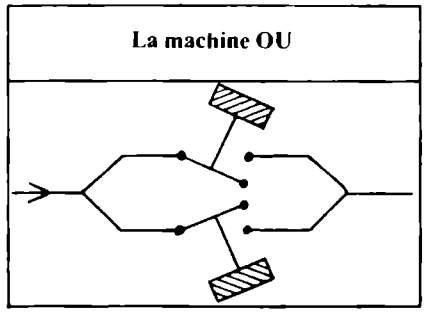
Notons que contrairement au robinet :

— quand un interrupteur est dit "ouvert", les électrons ne passent pas.
— quand un interrupteur est dit "fermé", les électrons passent.

La logique binaire

En associant deux transistors d'une façon judicieuse, il est possible de réaliser deux machines bien intéressantes :





La machine ET laisse passer le courant si l'interrupteur A ET l'interrupteur B sont fermés en même temps.
La machine OU laisse passer le courant si l'interrupteur A OU l'interrupteur B est fermé, de même si les deux interrupteurs A et B sont fermés en même temps. Dans ce cas, on parle d'une machine OU non exclusive.
Ici, nous attend une bonne surprise :
— *Toutes* les opérations possibles entre des séries de bits sont réalisables avec uniquement les deux opérations de base ET et OU et un "opérateur", appelé NON qui transforme les Ø en 1 et inversement. Un microprocesseur n'est donc qu'un gigantesque Meccano réalisé avec ces trois pièces de base associées à des fils transporteurs de bits.
Pour vous persuader de l'heureuse réalité de ce miracle, nous nous contenterons de vous dire qu'il est possible de construire un circuit qui effectue l'addition de deux chiffres binaires. Ce n'est pas grand-chose, mais vous devriez comprendre que cela suffit en pensant à ce qui se passe pour les calculs numériques :
• le calcul numérique le plus compliqué peut se réduire à des successions d'opérations élémentaires (+, ×, −);
• la multiplication peut se réduire à une suite d'additions et la division à une suite de soustractions;
• la soustraction n'est qu'une addition "à l'envers";
• l'addition la plus compliquée peut s'effectuer chiffre à chiffre.
Tout calcul numérique peut donc se réduire à une succession d'additions de deux chiffres; le problème le plus ennuyeux étant finalement celui de la "retenue".
Vous savez bien que cette cascade de réductions est aussi possible avec les suites de Ø et de 1 ; il suffirait donc de savoir effectuer l'addition de deux chiffres binaires pour savoir quasiment *tout* faire.

LE CODAGE BINAIRE DES INFORMATIONS

Notre explication est, en fait, fondée sur la compréhension de deux points essentiels :

1er point : Toute information peut s'exprimer (c'est-à-dire se coder) par une suite éventuellement très longue de Ø et de 1.

2e point : Les composants d'un microprocesseur ont pour mission de transformer les suites de Ø et de 1 en d'autres suites de Ø et de 1 exprimant une information de sortie, différente de l'information d'entrée.

L'unité d'information : le bit

Au départ, la matière première, c'est l'électricité, celle qui alimente les prises de courant. La première tâche du micro-ordinateur consiste à abaisser la tension courante (220 V) à 12 V continus grâce à un adaptateur. La seconde tâche consiste à rythmer le flux électrique par une horloge interne qui donne la cadence. L'horloge du TO7-7Ø bat à 1 mégahertz, c'est-à-dire 1 million de fois par seconde (1 MHz).

Le courant électrique peut alors être considéré comme une suite d'impulsions qui vont devenir par la magie des mots et la puissance de l'imagination des **unités élémentaires d'information,** en anglais "bit" *(binary digit)* aujourd'hui considéré comme un mot français.

Le bit a pour seule vertu de proposer une alternative : présence ou absence suivant que le courant passe ou ne passe pas. Mais cette seule concentration d'électrons en un endroit précis, malgré son apparente pauvreté, peut devenir, par les effets conjugués de l'accumulation des bits et d'un codage subtil, le support d'une information complexe. Le premier codage consiste à comprendre comme 1 la présence d'une impulsion électrique et comme Ø son absence.

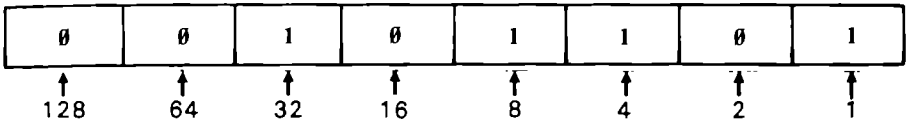
Le défi qu'il s'agit de relever est en définitive le suivant : toute information peut-elle s'exprimer par une suite de Ø et de 1 ?
Dans notre monde et pour l'usage que nous en faisons, l'information se présente le plus souvent sous trois formes : nombres, images et textes.

Nous allons montrer rapidement comment ces trois formes d'information peuvent être représentées par une série de Ø et de 1, c'est-à-dire sous forme binaire.

Les nombres

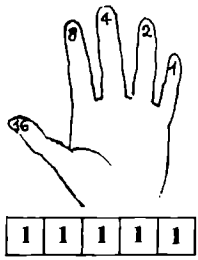
Il existe en informatique un système de représentation des nombres qui utilise les seuls chiffres Ø et 1. C'est le système *binaire.*
Il repose sur les mêmes principes que le système décimal que nous utilisons couramment (à cause de nos dix doigts...) avec cette seule différence que c'est le nombre 2 qui joue dans le système binaire le rôle de 10 dans le système décimal. D'ailleurs, en binaire, 2 s'écrit 1Ø!
Pour faciliter la manipulation des nombres binaires, les bits sont regroupés par série de 8, un **octet.**

Voici un octet :



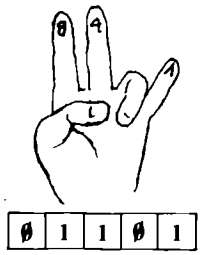
Dans un octet, la valeur d'un bit, qu'on appelle son *poids,* dépend de sa position dans l'octet.
- Le bit le plus à droite "compte pour" 1.
- Le bit à gauche de ce dernier compte pour 2.
- Le bit à gauche de ce dernier compte pour 4.
- Le bit à gauche de ce dernier compte pour 8

et ainsi de suite.
L'octet précédent représente donc le nombre :
32 + 8 + 4 + 1 = 45



Un nombre est écrit sur chaque doigt de cette main. Lorsqu'ils sont tous levés, elle représente le nombre 31 car :
16 + 8 + 4 + 2 + 1 = 31

Code binaire



Voici le nombre 13

Code binaire

Dans un ordinateur, la mémoire est un monstre pouvant disposer par exemple, de 48 mille mains à 8 doigts !

Les types classiques de nombres peuvent ainsi être codés à l'aide de quelques octets. Pour le TO7-7Ø, les entiers (de −32 768 à +32 767) sont représentés sur 2 octets, les réels en simple précision (8 chiffres) sur 4 octets et les réels en double précision (16 chiffres) sur 8 octets.

Les textes

Un texte étant composé de signes conventionnels, il suffit de coder chacun de ces signes par un nombre ce qui ne pose pas de problèmes particuliers si ce n'est celui d'une nécessaire standardisation internationale.
Un tel standard existe, c'est le code ASCII (American Standard Code for Information Interchange). Il est respecté par presque tous les constructeurs et développeurs. Chaque caractère est représenté sur un octet, ce qui offre 256 possibilités.
Le TO7-7Ø utilise les 128 premiers pour son usage propre et réserve les 128 autres pour les utilisateurs désireux de définir leurs propres caractères. Le code ASCII est organisé en tranches :

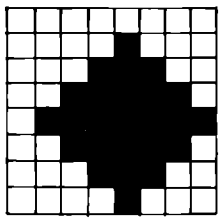
— les caractères alphabétiques majuscules de 65 à 90,
— les caractères alphabétiques minuscules de 97 à 122,
— les chiffres de 48 à 57.

Les codes de 1 à 32 concernent les caractères dits de contrôle. Ils permettent entre autres choses de représenter la mise en page des textes : passage à la ligne, tabulation, déplacement du curseur, clignotements, etc.
Les autres codes concernent les signes de ponctuation, signes mathématiques et caractères particuliers ($, £, à, etc.).

Les images

Pour ramener une image à une suite de Ø et de 1 ou plus précisément à une suite d'octets, il faut d'abord accepter de ramener l'image à un arrangement de points. Notons que de ce point de vue, l'informatique n'a rien inventé : les mosaïques, Seurat, la télévision, pour ne citer que quelques exemples de précurseurs, utilisent des arrangements de points.

Par exemple, le caractère "as de carreau" peut être codé sur une grille 8 × 8



codée Ø ou, en binaire : ØØØØØØØØ
codée 8 ou, en binaire : ØØØØ1ØØØ
codée 28 ou, en binaire : ØØØ111ØØ
codée 62 ou, en binaire : Ø0111110
codée127 ou, en binaire : Ø1111111
codée 62 ou, en binaire : Ø0111110
codée 28 ou, en binaire : ØØØ111ØØ
codée 8 ou, en binaire : ØØØØ1ØØØ

L'image, la plupart du temps, sera visible sur l'écran du téléviseur. Chaque point du quadrillage s'appelle un *pixel* (de l'anglais *picture element*). Suivant qu'il sera allumé ou éteint, le bit correspondant prendra la valeur 1 ou Ø.
A cette description d'une forme, on peut ajouter une description des couleurs en doublant le bit de chaque pixel de bits correspondant à une couleur. S'il y a 8 couleurs, c'est que les couleurs sont codés sur 3 bits et s'il y en a 16, c'est qu'elles sont codées sur 4 bits.

Le Basic, par exemple, donne accès à 16 couleurs :
— les 8 couleurs primaires par COLOR et SCREEN,
— les 8 couleurs pastel par une autre commande.
Ces couleurs sont obtenues par synthèse additive de trois couleurs fondamentales : bleu, vert et rouge.

Bleu	Vert	Rouge		Code
0	0	1	→ Rouge	1
0	1	0	→ Vert	2
0	1	1	→ Jaune	3
1	0	0	→ Bleu	4
1	0	1	→ Magenta	5
1	1	0	→ Cyan	6
1	1	1	→ Blanc	7

L'essentiel en définitive est de retenir qu'une image en couleurs peut être ramenée à une longue suite de Ø et de 1. A condition évidemment que cette suite puisse être comprise de la même façon par le micro-ordinateur et par l'homme qui l'utilise. On peut imaginer que plus le nombre de pixels sur l'écran est grand et plus les couleurs sont nombreuses, plus le code sera long mais plus l'image sera belle.

LE MICROPROCESSEUR ET LES MÉMOIRES

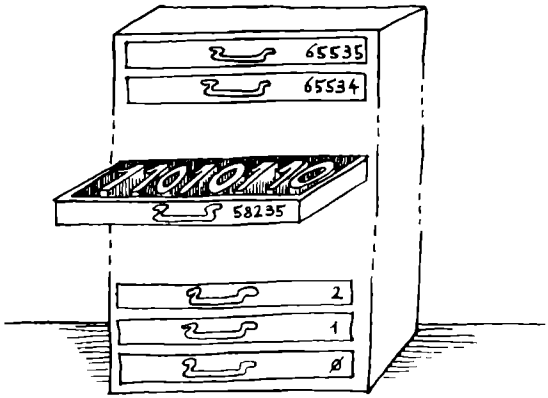
Les principes de codage étant admis, il reste à comprendre comment l'outil qu'est le micro-ordinateur est capable de remplir les devoirs que l'utilisateur est en droit d'en espérer. Concrètement, l'outil devra être capable de réaliser un certain nombre d'opérations sur cette matière première qu'est l'information :

— saisir,
— traiter,
— mémoriser,
— restituer.

Nous examinons dans cette partie les fonctions de traitement et de mémorisation.

Le microprocesseur

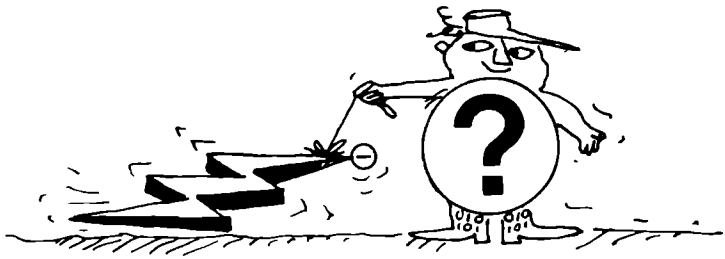
Le microprocesseur du TO7-7Ø est le 6809-E. Il s'agit d'un modèle 8 bits, c'est-à-dire qu'il manipule un octet à la fois. Son rôle est de superviser les échanges d'information dans l'unité centrale, d'effectuer certaines manipulations comme les opérations entre données numériques et d'aiguiller les échanges avec l'extérieur vers les circuits spécialisés.
C'est de très loin la puce la plus active de l'ensemble.
A l'intérieur du microprocesseur, les échanges se font par deux octets. Le microprocesseur dispose de six registres qui sont des mémoires internes permettant de stocker temporairement des données avant de les envoyer éventuellement vers des mémoires extérieures (voir plus loin). Il peut être commandé à l'aide de 59 instructions élémentaires.



La case-mémoire 58235 contient l'octet : 11Ø1Ø11Ø

Les mémoires

Concrètement, les mémoires sont des *puces*, c'est-à-dire des circuits intégrés. Sur la carte, rien ne les distingue du microprocesseur, si ce n'est la dimension : les puces-mémoire sont en général plus petites que le microprocesseur et leur architecture interne est elle aussi très particulière.
Le rôle des mémoires est très nettement celui d'assistants du microprocesseur. On peut se faire une idée assez juste d'une puce-mémoire, en imaginant une armoire comportant un très grand nombre de tiroirs, les cases-mémoire, dans chacun desquels est rangé un octet qui peut être lu ou modifié par le microprocesseur.
Pour être juste, il faudrait dire que les cases-mémoire sont réparties dans plusieurs armoires, c'est-à-dire plusieurs puces.
Quoi qu'il en soit, chaque case-mémoire est repérée par une "adresse", un nombre : la première case-mémoire a pour adresse Ø, la dernière 65535.
Le nombre de cases-mémoire ($65536 = 2^{16}$) n'a pas été fixé par une fantaisie du constructeur; il correspond aux possibilités du microprocesseur lui-même. Le 6809 note les adresses des mémoires avec lesquelles il communique, sur deux octets.
Il y a donc 16 bits permettant de retrouver l'adresse la plus basse, Ø, jusqu'à l'adresse la plus haute, 1111111111111111, soit 65535 en notation décimale.
On dit que la capacité d'adressage de ce microprocesseur est de 64 kilo-octets.
Si l'on y regarde d'un peu près, les mémoires sont amenées à rendre des services suffisamment divers pour que l'on ressente la nécessité d'une classification. Par bonheur, cette classification en recouvre une autre au niveau technologique.
Les différents types de mémoires se décrivent à l'aide de trois critères :
— la capacité,
— les possibilités de les lire et d'y écrire,
— le temps d'accès.
La capacité se mesure en kilo-octets qui équivalent à 1 024 octets. 1024 est la puissance de 2 la plus proche de 1000 ($2^{10} = 1\,024$).
Le temps d'accès correspond au temps mis par la mémoire pour restituer l'information qu'elle possède au microprocesseur dès l'instant où il connaît son emplacement.
A partir de ces caractéristiques, on comprendra mieux la classification des mémoires en trois catégories : mémoire morte, mémoire vive, mémoire de masse.



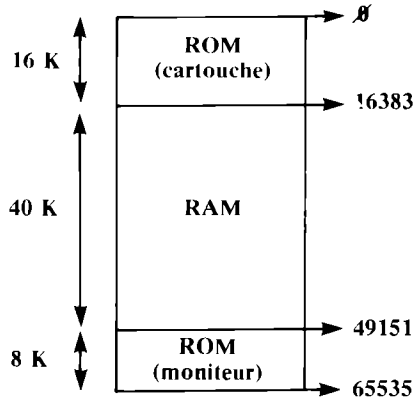
La mémoire morte

Abréviation MEM ou ROM (Read Only Memory).
La ROM du TO7-7Ø occupe 6 kilo-octets, soit 6 144 cases-mémoire. Son contenu est inaltérable même en l'absence d'alimentation électrique. Le temps d'accès est très court (quelques millisecondes).
La ROM contient le moniteur qui constitue une succession d'indications que le microprocesseur ira lire dès mise sous tension afin d'être opérationnel pour la suite de la manœuvre.
Ce programme écrit par les ingénieurs de chez Thomson permet à votre micro-ordinateur d'exécuter des tâches primordiales telles que :
— lire le clavier,
— afficher à l'écran,
— gérer le crayon optique,
— contrôler le lecteur de cassettes,
— émettre des sons, etc.

Il est constitué d'un certain nombre de routines spécialisées dans chacune de ces fonctions qui dialoguent directement avec le microprocesseur. Vous pouvez utiliser ces routines pour vos propres besoins à condition de connaître les adresses ROM où elles se trouvent (points d'entrée) et les protocoles d'utilisation. Un descriptif détaillé vous en est donné à la fin de ce manuel.
En raison de son rôle vital pour le fonctionnement du micro-ordinateur, on comprend mieux la nécessité d'implanter le moniteur dans une mémoire inaltérable où seule la lecture est possible.
Une autre ROM peut s'ajouter à celle-ci sous la forme d'une cartouche. Nous en parlerons plus loin.

La mémoire vive

Abréviation MEV ou RAM (Random Access Memory).
Tout autres sont les mémoires vives qui comme les mémoires mortes bénéficient d'un temps d'accès très court.
La RAM est mise à la disposition de l'utilisateur pour qu'il y stocke des informations personnelles. En l'absence d'alimentation électrique, la RAM est vide. En écrivant un programme Basic par exemple, on remplit progressivement la RAM des données définissant le programme. Si on éteint, tout disparaît.
Dans une mémoire vive, le microprocesseur peut aussi bien lire qu'écrire.
Pour donner à l'utilisateur un confort suffisant, la capacité de la RAM doit être assez importante. On considère que c'est une caractéristique très importante d'un micro-ordinateur.
La mémoire du TO7-7Ø se partage très naturellement en deux tranches : une tranche ROM et une tranche RAM.
La taille de la ROM garantit la puissance du système et la taille de la RAM garantit le confort de l'utilisateur.
Le schéma suivant est une carte mémoire *(memory map)* propre au TO7-7Ø. Il donne les adresses extrêmes de ROM et de RAM. On en trouvera une version plus détaillée page 102.



Les 8 premiers kilo-octets de la RAM sont réservés à la mémoire d'écran (voir page 56).

La mémoire de masse

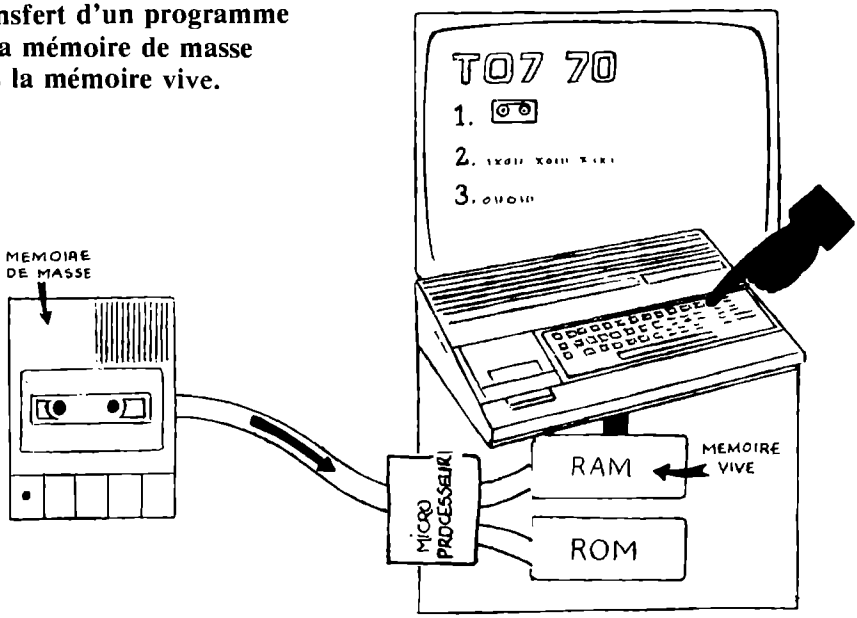
La volatilité de la RAM est évidemment l'un de ses défauts. Lorsque le micro-ordinateur n'est plus alimenté électriquement, son contenu disparaît complètement et instantanément. C'est très fâcheux s'il s'agit des résultats d'un long travail.
Il est donc absolument nécessaire de disposer en plus de la RAM résidente d'une mémoire de stockage supplémentaire qui offrirait les avantages suivants :
— forte capacité,
— non volatile,
— possibilité de stockage et de transport.
Il s'agit donc de disposer pour un faible prix d'un support de mémoire permanente de forte capacité disponible aussi bien pour la lecture que pour l'écriture. Deux techniques courantes permettent de réaliser ces objectifs :
— la cassette,
— la disquette.
On peut facilement stocker plusieurs dizaines de kilo-octets sur une cassette audio ordinaire et plusieurs centaines sur une disquette. On peut donc se créer une bibliothèque personnelle de programmes, soit achetés dans le commerce, soit résultant de son propre travail de création.
Ces avantages se paient au niveau du temps d'accès. L'inconvénient est particulièrement grave pour la cassette dont le défilement séquentiel devant la tête de lecture du magnétophone peut entraîner des temps d'accès de plusieurs minutes.
Le système disquette présente, par rapport au système cassette, plusieurs avantages : l'accès est direct et le temps d'accès se réduit à quelques dixièmes de seconde.

Remplir la mémoire

Remplir la mémoire du TO7-7Ø consiste très concrètement à placer un octet dans chacune des cases-mémoire.
Qui va le faire?
Essentiellement, deux types de personnes :
— le créateur de la machine,
— l'utilisateur de la machine.
Ou si l'on préfère :
— les ingénieurs de chez THOMSON,
— vous.
La ROM a été remplie une fois pour toutes par les concepteurs du TO7-7Ø. La RAM, en revanche, est à la disposition de l'utilisateur et sera remplie, partiellement le plus souvent, par lui.

Les octets contenus dans la RAM constituent généralement un programme et ses données. La manœuvre la plus simple pour une telle opération se fait le plus simplement par transfert d'une mémoire de masse vers la RAM. Dans le cas du TO7-7Ø connecté à un lecteur de cassettes, il suffira, après avoir branché le système, de taper l'option 1 du menu.

Transfert d'un programme de la mémoire de masse vers la mémoire vive.



Le moniteur (en ROM) est responsable de la saisie au clavier (touche 1). Mais il n'est pas responsable du transfert lui-même qui doit être pris en charge par un autre programme résidant en ROM sous la forme d'une cartouche, par exemple une cartouche-langage, dont le rôle est fondamental pour votre micro-ordinateur.

LA CARTOUCHE

Elle contient essentiellement des puces-mémoire de type ROM (mémoire morte) dont la capacité peut varier de 16 kilo-octets à 64 kilo-octets. Leur contenu va servir d'intermédiaire entre l'utilisateur et le groupe formé par le microprocesseur et ses mémoires. L'intérêt de la cartouche est que le programme qu'elle contient est immédiatement disponible et ne consomme pas d'espace dans la mémoire vive (RAM) qui reste entièrement disponible pour l'utilisateur.

La cartouche Basic

Elle contient, selon la version, 16 ou 32 kilo-octets de mémoire morte qui constituent ce qu'on appelle un **interpréteur BASIC.**
BASIC est un langage évolué car les instructions codées contenues dans la ROM de la cartouche vont permettre à l'utilisateur de donner des ordres au microprocesseur.
Ces ordres sont exprimés dans un langage simple. Ce langage est interprété (par le BASIC) pour le rendre intelligible au microprocesseur. S'il s'agit, par exemple, d'obtenir le résultat de l'opération 48 × 1Ø24, il ne sera pas nécessaire de demander au microprocesseur de ranger 48 et 1Ø24 à certaines adresses et le résultat à une autre, puis d'aller le chercher. Il suffira de taper au clavier :
PRINT 48 × 1Ø24
puis d'appuyer sur la touche ENTRÉE pour voir apparaître le résultat : 49 152.
On observera que BASIC n'est pas évolué au point d'avoir appris le français, mais c'est tout de même un beau progrès.
Pour pratiquer un jeu sur cassette ou sur disquette écrit en BASIC, il faut donc installer la cartouche BASIC dans la trappe pour permettre au microprocesseur d'interpréter les instructions du programme, puis de charger ce programme dans la RAM (mémoire vive) par l'intermédiaire du magnétophone ou du lecteur de disquettes.
Il existe d'autres langages de programmation susceptibles d'être enregistrés dans une cartouche : LOGO, FORTH, ASSEMBLEUR...

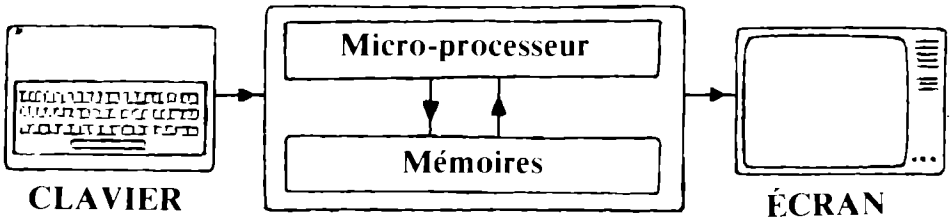
Les autres cartouches

S'il s'agit d'une cartouche de jeu, le principe d'installation et de fonctionnement est le même que pour la cartouche BASIC, à ceci près qu'avec cette cartouche vous ne pourrez que pratiquer ce jeu et renoncer à connaître le résultat de l'opération 48 × 1Ø24, ou à écrire un programme.
Les instructions contenues dans la mémoire d'une cartouche de jeu sont directement interprétables par le microprocesseur.
Ces cartouches sont écrites en langage assembleur du 68Ø9.

LES PÉRIPHÉRIQUES

Maintenant que le rôle fondamental de la cartouche enfichable a été démontré, il nous reste à revenir en arrière pour expliquer le rôle non moins fondamental de deux instruments qui permettent à l'utilisateur de donner ses instructions et d'en recevoir le résultat : le clavier et l'écran.

- Le clavier s'appelle un *périphérique d'entrée.*
- L'écran s'appelle un *périphérique de sortie* par son affichage et un *périphérique d'entrée* grâce au crayon optique.



Le clavier et le code ASCII

C'est le clavier qui va vous permettre de donner des instructions au microprocesseur; les touches sur lesquelles vous appuyez seront enregistrées et traduites par l'interpréteur sous forme de codes identifiables par le microprocesseur.
Bien entendu, pour le microprocesseur, comme pour tous les composants électroniques contenus dans le micro-ordinateur, seuls les signes Ø et 1 peuvent être identifiés.
Le clavier du TO7-7Ø, comme tous les autres, comporte bien d'autres signes, en particulier, les 26 lettres de l'alphabet.
Lorsque vous appuierez sur la lettre A, le micro-ordinateur ne pourra comprendre qu'il s'agit d'un A, que si ce signe est traduit dans un code numérique qui permet d'être exprimé avec des Ø et des 1.
Le code ASCII* a été mis au point pour coder tous les caractères courants par un nombre.
Chaque caractère est représenté par un octet, ce qui offre 256 possibilités.

Les 128 premiers codes sont imposés par le constructeur, les autres peuvent être mis à votre disposition (en Basic par exemple).

* American Standard Code for Information Interchange.

Voici le tableau complet du code ASCII.

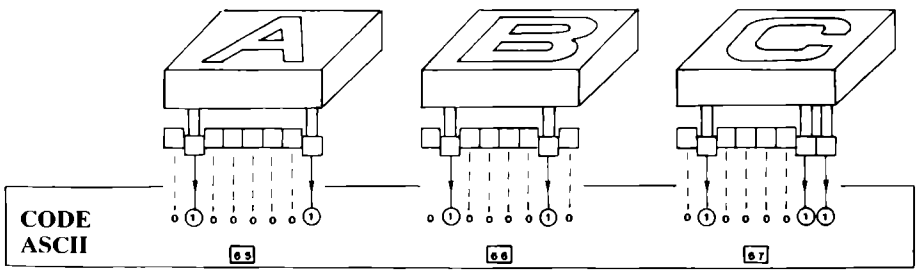
code	caractère
2	touche STOP
8	flèche ←
9	flèche →
1Ø	flèche ↓
11	flèche ↑
12	touche RAZ
13	touche ENTREE
22	touche ACC
28	touche INS
29	touche EFF
3Ø	touche ⊐
32	barre espacement
33	!
34	"
35	#
36	$
37	%
38	&
39	'
4Ø	(
41	)
42	*
43	+
44	,
45	-

46	.
47	/
48	Ø
49	1
5Ø	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
6Ø	<
61	=
62	>
63	?
64	a
65	A
66	B
67	C
68	D
69	E
7Ø	F
71	G
72	H
73	I

74	J
75	K
76	L
77	M
78	N
79	O
8Ø	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
9Ø	Z
91	[
92	\
93	]
94	^
95	_
96	`
97	a
98	b
99	c
1ØØ	d
1Ø1	e

1Ø2	f
1Ø3	g
1Ø4	h
1Ø5	i
1Ø6	j
1Ø7	k
1Ø8	l
1Ø9	m
11Ø	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
12Ø	x
121	y
122	z
123	{
124	|
125	}
126	~
127	■

Tout se passe comme si, sous chaque touche du clavier, se trouvaient 8 contacts électriques.
Lorsque vous enfoncez une touche, le courant passe dans certains fils et pas dans d'autres.
En fonction du code qu'il reçoit, le microprocesseur reconnaît qu'il s'agit d'un A ou d'une autre lettre.



Chaque caractère que vous frappez au clavier est interprété séparément des autres.
Ainsi, l'ordre PRINT est-il codé par une série de nombres : 8Ø, 82, 73, 78, 84.
Cette série de nombres est alors reconnue par l'interpréteur BASIC comme un mot du langage correspondant à une fonction précise (l'affichage), et aussitôt recodée par un seul nombre.

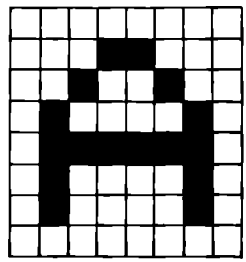
L'écran

Si tout se passe bien, lorsque vous appuyez sur la touche marquée A, cette lettre s'inscrit à l'écran.
Cette petite expérience ne soulève pas en général l'enthousiasme de l'utilisateur qui trouve cela à la fois banal et évident.
Pourtant, vous pouvez imaginer qu'il s'est passé des choses assez étonnantes entre l'appui de la touche A et son affichage sur l'écran.
En particulier, rien de ce qui peut ressembler à un A n'a circulé dans les fils du micro-ordinateur.
Très rapidement, ce A est devenu le nombre binaire Ø1ØØØØØ1 (65 dans le code ASCII).
Pour comprendre la suite du processus qui aboutira à l'affichage d'un A sur l'écran, il faut regarder du côté de votre poste de télévision.
La surface de l'écran est recouverte d'une matière phosphorescente qui devient lumineuse lorsqu'elle reçoit un faisceau d'électrons, une sorte de rayon électrique.
Le canon qui émet le faisceau d'électrons balaie l'écran ligne par ligne à une très grande vitesse : chaque seconde, l'écran est balayé 50 fois!
L'impact du faisceau crée une mosaïque de points lumineux ou sombres, chaque point s'appelle un pixel *(picture element)*. L'écran du TO7-7Ø contient 64000 pixels répartis sur une grille rectangulaire 320 × 200.

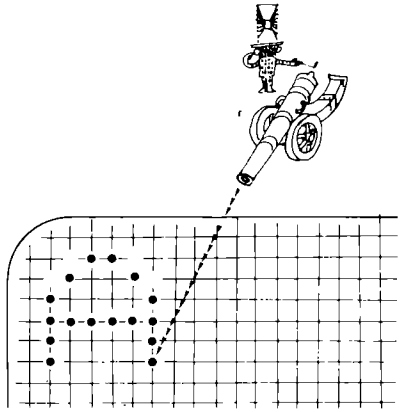
L'affichage d'une lettre

Revenons à la touche A sur laquelle vous avez appuyé et que le micro-ordinateur a ressenti comme le code Ø1ØØØØØ1.
A partir de cette simple information et tenant compte du fait qu'elle arrive du clavier, le microprocesseur va lire en mémoire morte une série de 8 octets qui vont lui permettre d'afficher la lettre A.

	Les 8 octets de la ROM	**Leur traduction graphique**
N° 1	Ø Ø Ø Ø Ø Ø Ø Ø	
N° 2	Ø Ø Ø 1 1 Ø Ø Ø	
N° 3	Ø Ø 1 Ø Ø 1 Ø Ø	
N° 4	Ø 1 Ø Ø Ø Ø 1 Ø	
N° 5	Ø 1 1 1 1 1 1 Ø	
N° 6	Ø 1 Ø Ø Ø Ø 1 Ø	
N° 7	Ø 1 Ø Ø Ø Ø 1 Ø	
N° 8	Ø Ø Ø Ø Ø Ø Ø Ø	



Pour faire parvenir cette information au canon à électrons, le microprocesseur les envoie par l'intermédiaire d'une puce spécialisée, *l'interface vidéo*, qui donnera clairement au canon l'ordre d'allumer certains points sur l'écran.



Il faut bien comprendre que l'affichage des touches frappées au clavier est un service rendu par le microprocesseur à l'utilisateur humain. Dès l'instant où vous avez appuyé sur la touche marquée A, le microprocesseur sait qu'il s'agit du caractère codé 65 et l'affichage écran est pour lui une perte de temps inutile. D'ailleurs, si vous n'allumez pas le téléviseur, le TO7-7Ø ne s'en trouvera absolument pas gêné. Il pourra continuer à recevoir vos ordres et à les exécuter imperturbablement. En allumant votre poste de télévision, vous retrouverez sur l'écran ce que vous aviez frappé. (Ne le faites pas, vous risqueriez d'endommager votre TO7-70.)

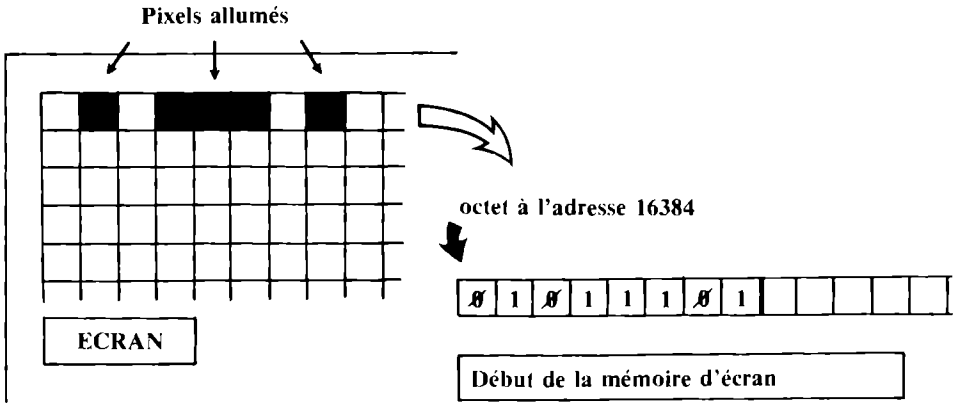
Le crayon optique

Le crayon optique est un dispositif sensible à la lumière créée par l'écran. Cette lumière qui vous paraît stable clignote en réalité 50 fois par seconde, soit trop vite pour que votre œil s'en rende compte. Par contre, le crayon optique perçoit très bien ce clignotement qui lui permet de connaître à quel endroit de l'écran vous le pointez.
Cette reconnaissance s'effectue par mesure du temps entre le début du balayage de l'écran et le moment où le crayon optique détecte l'impulsion de lumière. Par exemple si le crayon optique détecte une impulsion après la moitié du temps mis pour balayer l'écran, cela voudra dire que vous l'avez pointé sur la ligne horizontale du milieu de l'écran. Pour connaître sa position exacte sur cette ligne horizontale, il opérera de la même façon sur la ligne; s'il détecte la lumière après la moitié du temps nécessaire au balayage d'une ligne, après le début de la ligne, cela voudra dire que vous l'avez pointé au centre exact de l'écran.

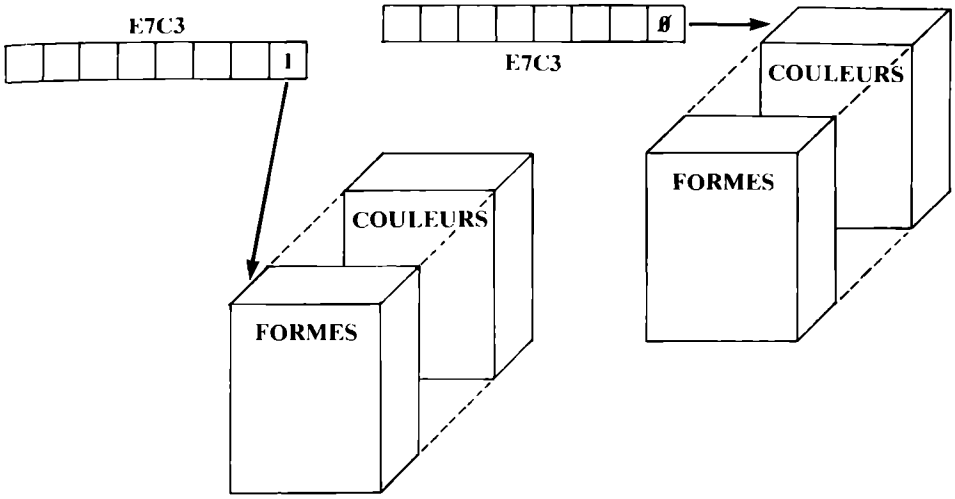
LA MÉMOIRE D'ÉCRAN

L'écran étant constitué d'un certain nombre de pixels (64 000), son état peut être considéré comme une suite de Ø et de 1 suivant qu'un pixel est allumé ou éteint : la mémoire d'écran est donc une copie conforme de l'écran. Le balayage se fait de haut en bas et de gauche à droite. Aux 64 000 pixels de l'écran sont donc associés 8 000 octets de mémoire vive.

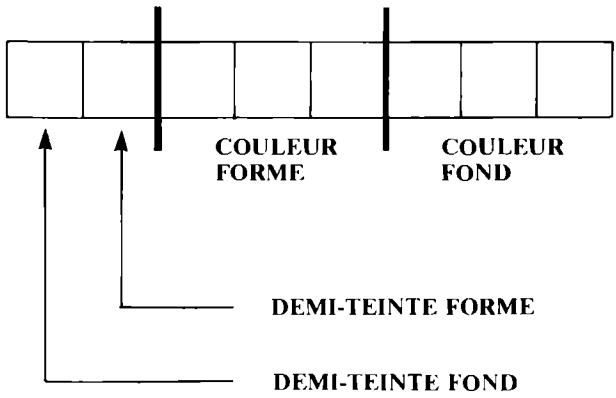
En ce qui concerne le TO7-7Ø et puisqu'il s'agit d'un micro-ordinateur couleur, il a fallu également prévoir une tranche de mémoire pour les couleurs. Ces deux tranches doivent évidemment être en relation étroite puisque chaque pixel est automatiquement associé à une couleur. Plutôt que de séparer ces deux tranches, les constructeurs ont préféré les associer par une méthode dite de commutation de banques-mémoire ou "bank switching" que nous allons décrire.
La zone mémoire comprise entre les adresses 16384 et 24575 (soit 8 192 octet dont les 192 derniers sont inutilisés) est une copie de l'écran. Une première banque correspond à la mémoire des formes. Chaque bit à Ø correspond à un point éteint, chaque bit à 1 correspond à un point allumé. La correspondance entre la forme de l'écran et la mémoire est très simple : les lignes d'écran sont parcourues de haut en bas et de gauche à droite.



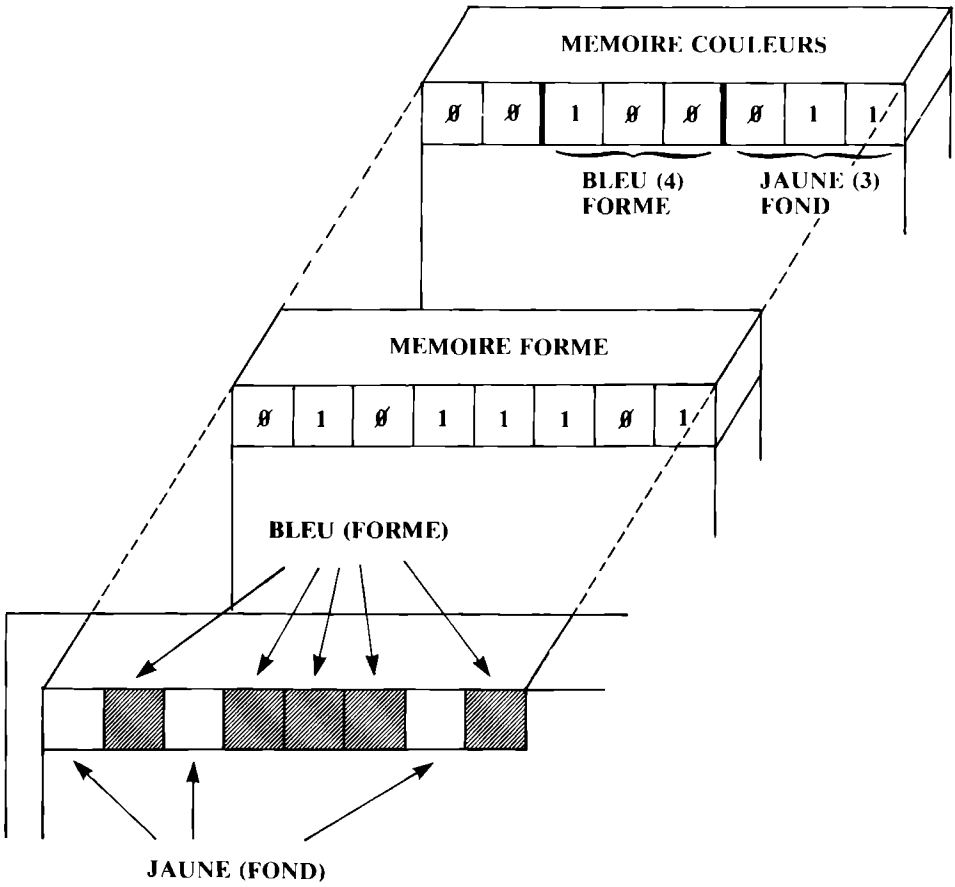
Le plus étonnant concerne la mémoire des couleurs qui se trouve implantée dans la même zone mémoire mais à des emplacements physiques différents. La mémoire des formes et la mémoire des couleurs peuvent être vues comme deux banques ayant les mêmes adresses et placées l'une derrière l'autre. Lorsque le microprocesseur recevra l'ordre d'aller lire à l'une de ces adresses, il ira dans l'une ou l'autre banque suivant la valeur prise par le bit de poids faible d'une case-mémoire particulière (à l'adresse E7C3).



Le dernier bit de cette mémoire sert de "drapeau" pour la lecture et l'écriture dans l'une des deux banques.
A chaque octet de la mémoire des formes correspond un octet dans la mémoire des couleurs mais la correspondance ne peut se faire au niveau du pixel car il est impossible de coder plus de deux couleurs sur un seul bit. Le principe retenu est le suivant : chaque groupe horizontal de 8 pixels peut recevoir 2 couleurs, une pour les points allumés (couleur de forme), une pour les points éteints (couleur de fond). Les 8 bits de chaque octet suffisent pour ce codage.
Leur organisation est la suivante :



La principale conséquence de cette organisation de la mémoire des couleurs est ce qu'on appelle la rigidité des couleurs.
Voici l'exemple d'un arrangement du premier octet de l'écran et sa traduction dans les deux banques de la mémoire d'écran.

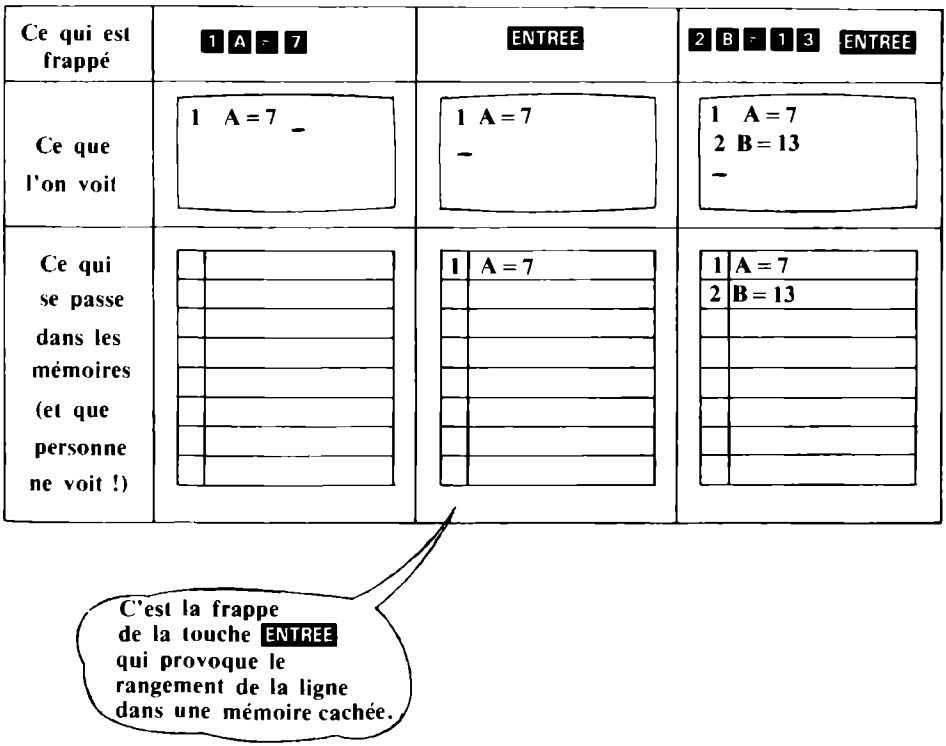


Signalons pour finir que c'est une méthode analogue de communication de banque mémoire qui permet d'étendre la capacité mémoire du TO7.7Ø de 64 k à 128 k.

QUE SE PASSE-T-IL SOUS LE CLAVIER DU TO7-70

Une fois la cartouche BASIC installée dans la trappe, vous pouvez écrire des programmes.
Voici une bande dessinée qui devrait vous permettre de comprendre le fonctionnement des mémoires mais surtout de bien distinguer entre le contenu de ces mémoires et ce qui est affiché à l'écran.
Il s'agit d'un court programme en BASIC qui n'utilise que des instructions simples.

- A = 7 pour ranger le nombre 7 dans la mémoire A,
- PRINT C pour afficher à l'écran le nombre contenu dans la mémoire C,
- RAZ pour effacer l'écran (équivalent de CLS),
- LIST pour rappeler le programme en mémoire,
- RUN pour lancer le programme,
- NEW pour vider le contenu de la mémoire vive.



ce qui est frappé	2 C = A * B ENTREE 4 P B I N T C ENTREE	RAZ	LIST ENTREE
ce que l'on voit	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C	-	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C
ce qui se passe dans les mémoires	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C

L'écran est vide mais le programme est bien conservé en mémoire

D'ailleurs l'ordre LIST le fait réapparaître sur l'écran

ce qui est frappé	RUN ENTREE
ce que l'on voit	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C RUN 91
ce qui se passe dans les mémoires	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C A 7 B 13 C 91

Entre l'appui sur ENTREE et l'affichage de 91, tout s'est passé très vite dans la mémoire. Voici un ralenti pour mieux comprendre

RALENTI

1	A = 7
2	B = 13
3	C = A*B
4	PRINT C
A	7

Exécution de l'instruction 1

1	A = 7
2	B = 13
3	C = A*B
4	PRINT C
A	7
B	13

Exécution de l'instruction 2

1	A = 7
2	B = 13
3	C = A*B
4	PRINT C
A	7
B	13
C	91

Exécution de l'instruction 3. Le microprocesseur a fait une multiplication

1	A = 7
2	B = 13
3	C = A*B
4	PRINT C
A	7
B	13
C	91

91

Exécution de l'instruction 4 : affichage du contenu de la mémoire C

SUITE

ce qui est frappé	NEW ENTREE	RAZ
ce que l'on voit	1 A = 7 2 B = 13 3 C = A*B 4 PRINT C NEW	–
Ce qui se passe dans les mémoires		

Le programme est affiché ainsi que les résultats mais les mémoires ont été vidées

L'écran et les mémoires ont été vidés. Nous voici revenus à notre point de départ

MICRO-DICTIONNAIRE

Adaptateur (de tension) [page 42] : Appareil électrique permettant de transformer le courant alternatif 220 volts en un courant continu de 12 volts. L'adaptateur de tension est intégré au TO7-7Ø.

Adresse (d'une case-mémoire) [page 47]: Numéro permettant de repérer chacune des 65 535 cases-mémoire.

ASCII (pages 44, 53) : American Standard Code for Information Interchange. Système de codage des *caractères*. La grille du code ASCII contient 128 caractères codés de Ø à 127 grâce à un seul *octet*.

ASSEMBLEUR (page 51) : Langage de programmation disponible sur une *cartouche* pour TO7-7Ø. Le langage assembleur très proche du langage du *microprocesseur* (langage machine) est d'un maniement délicat mais permet une exécution très rapide des programmes.

BASIC (page 51) : Langage de programmation disponible sur une *cartouche* pour TO7-7Ø. C'est le langage le plus utilisé en micro-informatique. C'est aussi l'un des plus faciles à apprendre.

Binaire *(système)* [page 42]: Système de numération à base 2, c'est-à-dire n'utilisant que les chiffres Ø et 1. C'est le système utilisé par le *microprocesseur.*

Bit (page 42) Plus petite unité d'information associée au passage ou à la présence d'un paquet *d'électrons.* Un bit représente 2 valeurs possibles : Ø et 1.

Canon à électrons (page 54): Appareil installé à l'arrière du téléviseur et qui permet de bombarder l'écran par un faisceau *d'électrons* afin d'afficher des images.

Capacité *(de la mémoire)* [page 47]: Nombre de *cases-mémoire adressables* par le *microprocesseur.* La capacité du TO7-7Ø est de 70 *kilo-octets.* Avec l'extension mémoire : 134 K.

Caractère (page 54) : Image codée sur une grille 8 × 8 grâce à laquelle il est possible d'afficher les lettres de l'alphabet, les chiffres et tous les signes présents au clavier.
Les *caractères de contrôle* (de Ø à 31 dans le code ASCII) gèrent le déplacement du curseur et le type d'affichage.

Cartouche (pages 20, 51) : Boîtier contenant de la *mémoire morte* et permettant à l'utilisateur de communiquer avec l'unité centrale. Il existe des cartouches de langage, de jeux ou de programmes utilitaires et créatifs (contient souvent 4 K, 8 K, 16 K mais peut atteindre 32 K ou 64 K).

Case-mémoire (page 47): Élément de la *mémoire* pouvant recevoir et conserver un *octet.*

Circuit intégré (on dit aussi **puce**) [page 38] : Plaque de *silicium* sur laquelle sont installés des composants électroniques *(transistors)* et des circuits de liaison *(bus).* L'intégration fait référence au nombre très élevé de composants.

Commutation de banques-mémoire (ou *bank switching*) [page 56] : Technique d'organisation de la mémoire qui permet d'affecter la même adresse à deux cases-mémoire physiquement distinctes.

Crayon optique (page 17) : Dispositif sensible à la lumière de l'écran qui permet d'effectuer des choix et de dessiner directement sur l'écran.

Électron (page 40) Particule élémentaire chargée électriquement. Les électrons sont la matière première de l'électricité.

FORTH (page 51) : Langage de programmation disponible sur *cartouche* TO7-7Ø. Forth est un langage récent (**Fourth**, quatrième génération) fondé sur la notion de pile.

Hertz (page 42) : Unité de fréquence mesurant le nombre d'oscillations de l'*horloge* en 1 seconde.
1 *mégahertz* vaut 1 000 000 hertz.

Horloge (page 42) : Dispositif utilisant l'oscillation régulière d'une pastille de quartz pour rythmer l'activité du *microprocesseur*. L'horloge du TO7-7Ø bat à la fréquence de 1 MHz (1 mégahertz).

Interface (page 54) : Circuit intégré assurant la communication entre l'unité centrale et un *périphérique*.
L'*interface vidéo* transmet les informations du microprocesseur au téléviseur.

Kilo-octet (page 47) : 1Ø24 *octets* ($1Ø24 = 2^{10}$).

LOGO (page 51) : Langage de programmation disponible sur cartouche TO7-7Ø. LOGO est un langage d'accès très simple, particulièrement utilisé avec les enfants.

Mémoire (pages 27, 47, 48, 49, 50, 56) : Circuit intégré assurant le stockage des informations.
Mémoire d'écran : tranche de mémoire chargée de conserver l'état de l'écran : points allumés, points éteints.
Mémoire de masse : mémoire extérieure au micro-ordinateur permettant un stockage magnétique permanent (cassettes, disquettes).
Mémoire morte (ROM) : mémoire réservée à la lecture. Son contenu est préservé en dehors de l'alimentation électrique.
Mémoire vive (RAM) : mémoire permettant la lecture aussi bien que l'écriture. Son contenu est effacé lorsque l'alimentation électrique n'est plus assurée. C'est la mémoire réservée à l'utilisateur.

Microprocesseur (pages 38, 46) : Circuit intégré assurant l'essentiel des calculs et manipulations d'informations nécessaires au fonctionnement du micro-ordinateur . Le microprocesseur du TO7-7Ø est le 6809-E.

Moniteur (page 48) : *2 sens :*
1° Programme stocké en *mémoire morte* permettant au *microprocesseur* de gérer la communication avec les *périphériques* dès la mise sous tension.
2° Écran vidéo spécialisé dans la communication avec un ordinateur.

Octet (page 43) : Groupe de 8 bits.

Périphérique (page 52): Appareil pouvant communiquer et servir l'unité centrale.
Le clavier est un périphérique d'entrée.
L'écran est un périphérique de sortie.
Autres périphériques : magnéto-cassettes, lecteur de disquettes, crayon optique, imprimante, manette de jeu, etc.

Pixel (page 54): Élément de l'image vidéo. L'écran du TO7-7Ø est affiché sur une grille 320 × 200, soit 64000 pixels.

Silicium (page 38) Métalloïde utilisé pour la fabrication des *circuits intégrés.*

Transistor (page 40) Composant électronique jouant le rôle d'un interrupteur ou d'un amplificateur.

Unité centrale : Groupe formé du *microprocesseur,* des *mémoires,* des *interfaces* et de l'ensemble des *bus.*

3.
LE MONITEUR DU TO7-70

SOMMAIRE

1.0 INTRODUCTION 69
1.1 Le moniteur du TO7-70 69
1.2 Accès aux routines du moniteur 70

2.0 GESTION DES INTERRUPTIONS 71

3.0 INITIALISATION 72
3.1 Test de démarrage à chaud ou à froid 72
3.2 Initialisation des PIA système jeux 72

4.0 GENERATEURS DE CARACTERES 74
4.1 Alphabet standard G0 74
4.2 Alphabet G2 74
4.3 Caractères utilisateur 74

5.0 PRIMITIVES DE GESTION D'ECRAN 75
5.1 Mise en mémoire couleur 76
5.2 Mise en mémoire caractère 76
5.3 Bip sonore 76
5.4 Gestion des caractères alphanumériques 77
5.4.1 Séquence normale 77
5.4.1.1 Codes affichables 77
5.4.1.2 Codes interprétables 77
5.4.1.3 Caractères utilisateur 78
5.4.2 Séquence de curseur ou de fenêtre 78
5.4.2.1 Positionnement du bas de la fenêtre 78
5.4.2.2 Positionnement du haut de la fenêtre 78
5.4.2.3 Positionnement du curseur en début de ligne 78
5.4.2.4 Positionnement du curseur 79
5.4.3 Séquence accent 79
5.4.4 Séquence TELETEL 80
5.4.5 Séquence "d'échappement" 81
5.4.5.1 Attributs de couleur 81
5.4.5.2 Attributs divers 82

6.0 CLAVIER 84
6.1 Lecture rapide du clavier 84
6.2 Décodage du clavier 84

7.0 PRIMITIVES GRAPHIQUES 86
7.1 Allumage ou extinction d'un point 86
7.1.1 Mode graphique 86
7.1.2 Mode caractère 87
7.2 Tracé d'un segment de droite 87
7.2.1 Mode graphique 87
7.2.2 Mode caractère 88
7.3 Segments horizontaux 88
7.4 Lecture de la couleur d'un point 88

8.0 LECTURE DE L'ECRAN 89
8.1 Caractère normal 89

8.2 Minuscule accentuée ou ç 89

9.0 GENERATION DE MUSIQUE 90

10.0 LECTURE DES MANETTES DE JEU 92

11.0 CRAYON OPTIQUE 93
11.1 Test du bouton du crayon optique 93
11.2 Lecture de la zone pointée 93

12.0 GESTION DE L'INTERFACE DE COMMUNICATION 94

13.0 GESTION DU LECTEUR-ENREGISTREUR DE CASSETTES 96

14.0 CONTROLEUR DE DISQUETTES 97
14.1 Entrées/Sorties disque physiques 97
14.2 Le format Basic Microsoft(R) 97
14.2.1 La table d'allocation des fichiers 99
14.2.2 Le catalogue 100

15.0 INFORMATIONS COMPLEMENTAIRES 102
15.1 Organisation de la mémoire 102
15.2 Points d'entrée standard du moniteur 104
15.3 Registres du moniteur 105
15.4 Adresses d'Entrées/Sorties 108
15.4.1 PIA système 108
15.4.2 PIA jeux 109
15.4.3 Interface de communication 109

1.0 INTRODUCTION

1.1 Le moniteur du TO7-70

Le moniteur du **TO7-70** est formé de différentes "routines" (programmes) ayant chacune une fonction bien précise: gestion de l'écran, du clavier, lecture du crayon optique et des manettes de jeu, génération de musique, gestion de l'interface de communication, du lecteur-enregistreur de programmes, etc...

Les programmes d'application en ROM, comme le Basic ou l'Assembleur, utilisent ces routines pour leurs propres besoins et vous pouvez faire de même. Le principe est le suivant:

1) Vous chargez avec des valeurs ayant une signification précise certains registres du 6809, et éventuellement certains registres en RAM (mots-mémoire sur un ou deux octets): ce sont les paramètres d'entrée.

2) Vous appelez la routine du moniteur qui effectue l'opération demandée, en utilisant le mécanisme explicité au § 1.2

3) Vous récupérez, s'il y a lieu, le résultat de l'opération dans certains registres du 6809, et/ou dans des registres en RAM: ce sont les paramètres de retour.

Dans certains cas, il n'y a pas de paramètres d'entrée, ou pas de paramètres de retour, mais le schéma général est toujours le même.

Dans ce qui suit, certains nombres ou adresses-mémoire seront écrits en hexadécimal pour plus de commodité : dans ce cas ils seront suivis par la lettre "H". Sauf avis contraire, tout nombre non suivi par un "H" est à lire en décimal.

Exemple: 10H = 16, 16H = 22, etc...

Les registres en RAM sur deux octets seront notés de la façon suivante: XXXXH-XXXXH où X représente un chiffre hexadécimal.

Exemple: mettre la valeur 4B0H dans le registre 605AH-605BH signifie: mettre la valeur 04 dans le mot-mémoire d'adresse 605AH et mettre la valeur B0H dans le mot-mémoire d'adresse 605BH.

Les nombres en binaire seront précédés du symbole "%". Les bits notés bi sont numerotés de gauche à droite, le bit de poids le plus faible étant le bit 0, et celui de poids le plus fort étant le bit 7.

Exemple: %11001100

b7 = 1

b0 = 0

Enfin, les bits ou chiffres hexadécimaux pouvant prendre une valeur quelconque seront notés respectivement "X" ou "x".

1.2 Accès aux routines du moniteur

L'accès normalisé à une routine du moniteur se fait en utilisant l'instruction "JSR" ou "JMP" suivie de l'adresse du point d'entrée. Ces adresses sont au nombre de 17.

EXEMPLE

```
PROG1   EQU     *
        LDB     #07
        JSR     $E803
        Instruction suivante  1
```

Le bip sonore se fera entendre, puis "instruction suivante 1" sera exécutée.

Mais si l'on rencontre dans un programme:

```
PROG1   EQU     *
        JSR     PROG2
        Instruction suivante  1
        . . . . . .

PROG2   EQU     *
        LDB     #07
        JMP     $E803
        Instruction suivante  2
```

Le bip sonore se fera entendre, puis "instruction suivante 1" sera exécutée, et non "instruction suivante 2".

Les sous-programmes du moniteur sauvegardent les registres du 6809. Au retour, tous les registres sont remis dans l'état qu'ils avaient lors de l'appel, sauf le registre code condition du 6809 et bien sûr les registres devant contenir des paramètres de retour (le plus souvent B, X et Y).

Il vous est FORTEMENT CONSEILLE d'appeler les routines du moniteur en utilisant l'accès normalisé. En effet, ces routines ne peuvent fonctionner correctement qu'avec le pointeur de pile et certains registres du 6809 dans un état bien précis.

La liste des codes des routines du moniteur est résumée dans le § 15.3.

2.0 GESTION DES INTERRUPTIONS

Les interruptions TIMER, IRQ, FIRQ, SWI et NMI du 6809 sont programmables, c'est-à-dire qu'à chacune correspond un registre en RAM contenant l'adresse du programme qui doit les traiter. A la mise sous tension ou après un redémarrage "à chaud", ces registres ont été initialisés avec l'adresse d'un programme du moniteur.

Le moniteur utilise les interruptions TIMER, initialisées à 100 millisecondes, pour gérer le clignotement du curseur et la répétition du clavier. Ces interruptions peuvent être aiguillées sur vos propres programmes en mettant à 1 le bit 5 du registre STATUS (6019H), et en mettant dans le registre TIMEPT (6027H-6028H) l'adresse de votre sous-programme de traitement. Votre sous-programme de traitement doit se finir obligatoirement par un JMP KBIN$ (E830H), ceci afin de valider l'interruption.

Pour gérer les IRQ, vous devez mettre l'adresse de votre sous-programme en IRQPT (6021H-6022H).

Pour gérer les FIRQ, vous devez mettre l'adresse de votre sous-programme en FIRQPT (6023H-6024H).

Pour gérer les SWI, vous devez mettre l'adresse de votre sous-programme en SWI1 (602FH-6030H). SWI2 saute directement en 6800H, et SWI3 en 7000H.

Pour gérer les NMI, vous devez mettre l'adresse de votre sous-programme en NMIPT (6025H-6026H).

Attention cependant: certaines routines du moniteur sont interruptibles, et vos programmes ne doivent pas modifier leurs paramètres.

Un JMP MENU$ (E82DH) vous fait revenir à la page d'en-tête.

3.0 INITIALISATION

A la mise sous tension ou après un "reset", le moniteur éxécute un certain nombre de tests et d'initialisations:

3.1 Test de démarrage à chaud ou à froid

Ce test est effectué pour ne pas modifier, en cas de redémarrage:

* le réglage du crayon optique

* le mot de code de mise en mode graphique, spécifique de l'imprimante utilisée.

Dans les deux cas:

Les registres d'aiguillage d'interruptions sont initialisés avec les adresses des routines moniteur correspondantes.

Les registres contenant les adresses des différentes tables (table de décodage du clavier, générateur de caractères standard, générateur de caractères utilisateur) sont réinitialisés de manière à pointer sur les tables standard.

Tous les autres registres sont remis aux valeurs standard, le plus souvent à zéro.

3.2 Initialisation des PIA système et jeux

Le PIA système 6846 est initialisé comme suit:

* registre de contrôle :

  - sortie : son
    écriture cassette

* registre de données :

  - entrée: lecture K7
    interrupteur crayon optique

  - sortie: sélection mémoires écran
    led clavier
    couleur tour écran

* timer :

  - Initialisé à 100 millisecondes

Le PIA système 6821 est initialisé comme suit:

* port A:

  - entrée: lecture matrice clavier

* port B:

  - entrée: sortie clavier

  - sortie: écriture matrice clavier
    sélection des banques

* registres de contrôle:

  - sortie: commande d'inscrustation
    commande moteur cassette

Les ports de données du PIA jeux sont initialisés en entrée.

4.0 GENERATEURS DE CARACTERES

4.1 Alphabet standard G0

Le générateur de caractères G0 est une suite de caractères affichables, correspondant au standard ASCII.

Chaque caractère est composé de 8 octets, qui forment une matrice de 8x8 points représentant le caractère. Le premier octet du caractère correspond a la ligne inférieure de la matrice.

Exemple: matrice definissant le "A"

```
0 0 0 0 0 0 0 0
0 0 0 1 1 0 0 0
0 0 1 0 0 1 0 0
0 1 0 0 0 0 1 0
0 1 1 1 1 1 1 0
0 1 0 0 0 0 1 0
0 1 0 0 0 0 1 0
0 0 0 0 0 0 0 0
```

Dans le générateur, A sera donc défini par la liste des octets suivants: 00H, 42H, 42H, 7EH, 42H, 24H, 18H, 00H

Dans le **TO7-70**, l'adresse du début du générateur G0 est contenue en RAM dans le registre PTGENE (60CFH-60D0H). Vous pouvez redéfinir l'alphabet standard en créant votre propre générateur selon le principe énoncé ci-dessus, puis en chargeant le registre PTGENE avec l'adresse de ce générateur.

4.2 Alphabet G2

L'alphabet G2 contient 20 caractères, dont les accents aigu, grave et circonflexe, le tréma et la cédille. Ces caractères sont définis suivant le principe décrit ci-dessus, et pour les accents, destinés à être combinés avec des caractères minuscules.

4.3 Caractères utilisateur

Outre le registre contenant l'adresse du générateur standard, un registre USERAF (602DH-602EH) est destiné à contenir l'adresse de début d'un générateur de caractères utilisateur. Ce générateur sera organisé comme expliqué ci-dessus, 8 octets étant nécéssaires pour définir un caractère. Quant aux codes, ils seront pris séquentiellement à partir de 80H. Cela veut dire que le code 80H correspondra au caractère défini par les 8 premiers octets, le code 81H au caractère défini par les 8 octets suivants, et ainsi de suite jusqu'au code FFH correspondant au 8 derniers octets du générateur. Vous pouvez ainsi définir jusqu'à 128 caractères en plus de l'alphabet standard.

5.0 PRIMITIVES DE GESTION D'ECRAN

Le TO7-70 peut gérer un écran pleine page de 320x200 points, avec 16 couleurs de forme, 16 couleurs de fond, et 16 couleurs de tour. En mode alphanumérique, il peut afficher 25 lignes de 40 caractères, définis par une matrice de 8x8 points, et ce dans une couleur au choix parmi 16, quelle que soit la couleur du fond. En mode graphique, l'unité centrale contrôle 8000 segments de 8 points chacun, et à chaque segment correspondent deux octets, situés à la même adresse logique. L'un est situé en mémoire dite "caractère", l'autre en mémoire dite "couleur", un bit de PIA permettant d'adresser physiquement l'une ou l'autre mémoire.

Dans un segment, un point peut prendre au choix deux couleurs: une couleur dite de "forme", auquel cas le bit correspondant dans l'octet en mémoire caractère est à 1, et un bit dit de "fond", auquel cas le bit correspondant en mémoire caractère est à 0. L'octet se trouvant à la même adresse mais en mémoire "couleur" complète cette information: les 3 bits de poids faible donnent la couleur du "fond", c'est-à-dire la couleur que devront prendre les points mis à 0, et les 3 bits suivants donnent la couleur de la "forme", c'est-à-dire la couleur que devront prendre les points mis à 1. Les 3 bits donnant la couleur sont BVR où B, V, R indiquent la présence de bleu, de vert, ou de rouge (les trois couleurs de base pour la T.V.) dans la couleur. Les 2 bits de poids fort servent à sélecter la couleur pastel en fond ou en forme. Le bit 6 mis à 0 sélecte les couleurs pastels pour la forme, et le bit 7 à 0 pour le fond.

Il est donc parfois imprécis de parler de points "allumés" ou "éteints": on peut avoir des points clairs appartenant au "fond" (donc de codage 0) et des points foncés appartenant à la "forme" (donc de codage 1).

Pour mémoire, sachez que:

Rouge + vert = jaune

Rouge + bleu = magenta

Bleu + Vert = cyan

Rouge + vert + bleu = blanc

et que bien sûr, ni rouge, ni bleu, ni vert donne du noir.

Exemple: lxxxx000

R = 0: pas de rouge

V = 0: pas de vert

B = 0: pas de bleu

p = 1: couleur sombre

Le résultat est un fond noir.

Autre exemple: xO111xxx

R = 1: rouge présent

V = 1: vert présent

B = 1: bleu présent

p = 0: couleur claire

Le résultat devrait être une forme "blanc clair". Cette couleur faisant double emploi avec le blanc, une exception à été faite: c'est l'orange.

Le **TO7-70** offre également la possibilité d'incrustation de l'image T.V. dans l'image écran: si votre **TO7-70** est équipé de la carte d'incrustation, la mise en mode "incrustation" (expliquée au § 5.4.4) laissera apparaître l'image T.V. à la place de toutes les zones noires, comme si la couleur noire devenait transparente.

5.1 Mise en mémoire couleur

* il suffit de mettre à 0 le bit 0 de l'adresse E7C3H, et ce bit là seulement.

Cette écriture vous permet d'aller sélectionner la mémoire couleur: tout ce que vous écrirez entre l'adresse 4000H et l'adresse 5FFFH modifiera la couleur des points correspondants.

5.2 Mise en mémoire caractère

* il suffit de mettre à 1 le bit 0 de l'adresse E7C3H, et ce bit là seulement.

Cette écriture vous permet d'aller sélectionner la mémoire caractère: tout ce que vous écrirez entre l'adresse 4000H et l'adresse 5FFFH modifiera l'état (fond ou forme) des points correspondants.

5.3 Bip sonore

* Envoi du caractère 07 à la routine PUTC$ (E803H).

* paramètre d'entrée:

- registre BUZZ (6073H)

Sur le **TO7-70** cette routine teste l'état du registre BUZZ. S'il est à 0, elle génère le bip sonore, sinon elle retourne sans rien faire.

5.4 Gestion des caractères alphanumériques

* adresse du point d'entrée: PUTC$ (E803H)

* paramètres d'entrée:

- registre 6809 B

5.4.1 Séquence normale

5.4.1.1 Codes affichables

Les codes affichables sont compris entre 20H et 7FH. Le caractère est affiché à la position courante du curseur. Si vous avez modifié le contenu du pointeur sur le générateur de caractères standard, les caractères affichés seront ceux que vous avez définis (entre 20H et 7FH).

5.4.1.2 Codes interprétables

Les codes interprétables sont compris entre 07 et 1FH. A la réception du code, l'une des opérations suivantes est effectuée:

- 07 : BEL--> "Bip" sonore.
- 08 : BS --> Déplacement d'une position vers la gauche ou recopie à gauche du caractère courant, si le registre COPCHR (6043H) contient la valeur FFH.
- 09 : HT --> Déplacement d'une position vers la droite ou recopie à droite du caractère courant, si le registre COPCHR (6043H) contient la valeur FFH.
- 0AH : LF --> Descente d'une ligne.
- 0BH : VT --> Remontée d'une ligne.
- 0CH : FF --> Effacement de toute la fenêtre.
- 0DH : CR --> Retour au début de la ligne courante.
- 0EH : SO --> Passage en mode TELETEL.
- 0FH : SI --> Retour au mode normal.
- 10H : DLE--> Rien.
- 11H : DC1--> Allumage du curseur.
- 12H : DC2--> Répétition du dernier caractère ASCII affiché sous la forme DC2/n où n est le nombre de répétitions.
- 13H : DC3--> Rien.
- 14H : DC4--> Extinction du curseur.
- 15H : NAK--> Rien.
- 16H : ACC--> Séquence caractère du G2.
- 17H : ETB--> Rien.
- 18H : CAN--> Effacement de la fin de la ligne à partir de la position courante du curseur.
- 19H : EM --> Rien.
- 1AH : SUB--> Rien.
- 1BH : ESC--> Séquence "d'échappement".
- 1CH : FS --> Rien.
- 1DH : GS --> Rien.
- 1EH : RS --> Retour du curseur dans le coin supérieur gauche de la fenêtre courante.
- 1FH : US --> Séquence de positionnement curseur ou de définition de fenêtre.

5.4.1.3 Caractères utilisateur

Les codes compris entre 80H et FFH sont des caractères utilisateur. Avant d'appeler la routine pour faire afficher le caractère correspondant au code, vous devrez avoir chargé le registre USERAF (602DH-602EH) avec l'adresse de votre générateur de caractères, comme cela est expliqué au § 4.3

5.4.2 **Séquence de curseur ou de fenêtre**

Le code US (1FH) définit une séquence de positionnement du curseur ou de définition de la fenêtre. Trois appels à la routine sont nécessaires, et peuvent être schématisés comme suit: US/N1/N2

5.4.2.1 **Positionnement du bas de la fenêtre (N1,N2 C <10H,19H>)**

Cette séquence définit le bas de la fenêtre courante, où le numéro de ligne est égal à 10 x n1 + n2, n1 et n2 (chiffre des dizaines et chiffre des unités) étant respectivement les quatre bits de poids faible de N1 et N2.

EXEMPLE:

Pour définir le bas de la fenêtre en ligne 21, on doit envoyer dans B:

1° appel: code 1FH dans B

2° appel: code 12H dans B

3° appel: code 11H dans B

5.4.2.2 **Positionnement du haut de la fenêtre (N1,N2 C <20H,29H>)**

Cette séquence définit le haut de la fenêtre courante, où le numéro de ligne est égal à 10 x n1 + n2, n1 et n2 étant respectivement les quatre bits de poids faible de N1 et N2.

5.4.2.3 Positionnement du curseur en début de ligne (N1,N2 C <30H,39H>)

Cette séquence positionne le curseur au début de la ligne de numéro 10 x n1 + n2, n1 et n2 étant respectivement les quatre bits de poids faible de N1 et N2.

5.4.2.4 Positionnement du curseur (N1,N2 C <40H,7FH>)

Le curseur est positionné en ligne n1, colonne n2, où n1 et n2 sont respectivement les 6 bits de poids faible de N1 et N2, avec : n1 C <0,24> et n2 C <1,40>.

EXEMPLE:

Pour positionner le curseur en ligne 3, colonne 20, il faudra mettre successivement dans B avant chaque appel:

- 1° appel: code 1FH dans B
- 2° appel: code 43H dans B
- 3° appel: code 54H dans B

5.4.3 **Séquence accent**

La séquence ACC est utilisée pour permettre l'affichage d'une minuscule accentuée ou d'un caractère du G2. On peut schématiser cette séquence comme suit: ACC/CODE/L.

Envoi d'un caractère accentué :

1° appel: envoyer ACC, code d'une séquence accent (16H)

2° appel: envoyer CODE qui est le code de l'accent :

41H pour l'accent aigu

42H pour l'accent grave

43H pour l'accent circonflexe

48H pour le tréma

4BH pour la cédille

3° appel: envoyer L qui est le code de la minuscule à accentuer. Si ce code ne correspond pas à un code de minuscule, alors la lettre dont le code est L est affichée, et l'accent est éffacé.

Envoi d'un caractère du G2 :

1° appel : envoyer ACC (16H), code d'une séquence G2

2° appel : envoyer CODE qui est le code du caractère :

23H pour la livre Sterling

24H pour le dollar

26H pour le dièse

2CH pour la flèche à gauche

2DH pour la flèche en haut

2EH pour la flèche à droite

2FH pour la flèche en bas

30H pour le degré

31H pour le signe plus-ou-moins

38H pour le signe division entière

3CH pour le signe 1/4

3DH pour le signe 1/2

3EH pour le signe 3/4

6AH pour le OE

7AH pour le oe

5.4.4 Séquence TELETEL

Dans ce mode les caractères envoyés perdent leur signification ASCII au profit du semi-graphique TELETEL.

Le semi-graphique divise le caractère standard en 6 zones :

```
   0 1 2 3   4 5 6 7 8
0
1   zone 0     zone 1
2
3   zone 2     zone 3
4
5
6   zone 4     zone 6
7
```

Chacune de ces zones est contrôlée par un bit. Si le bit est à 1 la zone prend la couleur forme, s'il est à 0, elle prend la couleur fond.

```
        0 x 1 x x x x x
bits      6   4 3 2 1 0
```

Les bits non-accessibles ci-dessus doivent toujours être dans l'état spécifié.

5.4.5 Séquence "d'échappement"

Cette séquence est utilisée pour positionner les attributs de couleur de l'écran, et d'autres attributs de la vidéo comme la taille des caractères, la mise en mode incrustation pour les TO7-70 équipés de la carte correspondante, le type de scroll, etc... Ces attributs peuvent être de deux types: courant ou plein écran. Dans le cas d'attributs de type courant, la séquence est de la forme: ESC/ATT, et dans le cas d'attributs de type plein écran, la séquence est de la forme: ESC/≠/ESPACE/ATT où ESC (1BH) est le code de la séquence d'échappement, ≠ le code du ≠ (23H), ESPACE le code de la barre d'espacement (20H), et ATT le code de l'attribut. Les codes d'attributs sont identiques, qu'ils soient de type courant ou plein écran, mais certains attributs (couleur du tour, attributs divers sauf la vidéo inverse) n'ont pas de signification en plein écran, et sont alors ignorés.

5.4.5.1 Attributs de couleur

ATT C <40H,47H> ou <50H,57H> ou <60H,67H> : La zone de l'écran qui sera modifiée est sélectionnée par les 4 bits de poids fort de l'octet "ATT", tandis que les 4 bits de poids faibles sélectionnent l'une des 8 couleurs primaires.

* 4 bits de poids fort:

  4 --> modification de la couleur de la forme.

  5 --> modification de la couleur du fond.

  6 --> modification de la couleur du tour.

* 4 bits de poids faible:

  00 --> noir

  01 --> rouge

  02 --> vert

  03 --> jaune

  04 --> bleu

  05 --> magenta

  06 --> cyan

  07 --> blanc

Cette table est etendue <70H,87H> pour donner accès aux 8 couleurs pastels. Les codes <70H,77H> affectent la couleur forme; les codes <78H,7FH> affectent le fond; et les codes <80H,87H> affectent le tour. Les valeurs qui suivent doivent donc être ajoutées à 70H pour la forme, à 78H pour le fond, et à 80H pour le tour:

00 --> gris

01 --> rose

02 --> vert clair

03 --> sable

04 --> bleu clair

05 --> parme

06 --> bleu ciel

07 --> orange

EXEMPLE:

Après la séquence suivante, vous obtiendrez un fond vert :

```
PUTC$   EQU     $E803
        LDB     #$1B    code ESC
        JSR     PUTC$
        LDB     #$5F    5 pour le fond, 3 pour le vert
        JSR     PUTC$
```

Et vous obtiendriez un tour orange avec :

```
        LDB     #$1B    code ESC
        JSR     PUTC$
        LDB     #$87    8 : tour pastel, 7 : orange
        JSR     PUTC$
```

5.4.5.2 Attributs divers

ATT C <4CH,4FH> ou <58H,5FH> ou <68H,6EH>

4CH --> caractères en taille normale.

4DH --> double hauteur, largeur normale.

4EH --> double largeur, hauteur normale.

4FH --> double taille.

58H --> masquage.

5FH --> démasquage.

5CH --> inversion de la vidéo.

68H --> écriture du caractère sans modifier la couleur.

69H --> écriture du caractère dans la couleur courante.

6CH --> suppression de l'incrustation

6DH --> mise en mode incrustation

6AH --> scroll à vitesse normale.

6BH --> mode page (pas de scroll).

6EH --> scroll doux.

Si un code supérieur à 6EH est envoyé, rien ne se passera. Si l'on envoie un code inferieur à 40H, rien ne se passera; même chose si l'on envoie dans une séquence plein écran le code d'un attribut qui n'a de sens qu'en attribut courant.

L'attribut de masquage consiste à écrire les caractères suivants en noir/noir jusqu'à ce que l'attribut de démasquage soit utilisé. Si l'attribut de démasquage est utilisé en plein écran, tous les caractères écrits en noir/noir seront visualisés dans la couleur utilisée lors du masquage.

En mode page, les codes 0CH, 1EH et 1FH réinitialisent les attributs de couleurs et de taille à moins que ceux-ci n'aient été définis en plein écran. Il en est de même pour la couleur, le masquage et la taille à la fin de chaque ligne à moins qu'ils n'aient été définis en plein écran.

EXEMPLE:

Apres la séquence suivante, les couleurs de la partie d'écran selectionnée seront inversées.

```
PUTC$   EQU     $E803
        LDB     #$1B    code de la séquence d'échappement
        JSR     PUTC$
        LDB     #$23    premier code de pleine page
        JSR     PUTC$
        LDB     #$20    deuxième code de pleine page
        JSR     PUTC$
        LDB     #$5C    code d'inversion vidéo
        JSR     PUTC$
```

6.0 CLAVIER

6.1 Lecture rapide du clavier

* adresse du point d'entrée: KTST$ (E809H)

* paramètres de retour:

- registres 6809 CC

Cette routine effectue une lecture rapide du clavier, pour tester si une touche a été enfoncée ou non. Si aucune touche n'a été enfoncée, le bit C du registre CC est mis à 0, sinon il est mis à 1.

6.2 Décodage du clavier

* adresse du point d'entrée: GETC$ (E806H)

* paramètres d'entrée:

- registres PTCLAV (60CDH-60CEH), BUZZ (6073H)

* paramètres de retour:

- registres 6809 B et CC

Cette routine effectue le décodage du clavier, et gère la répétition d'une touche après un certain délai. Sur le TO7-70, le bip sonore qui se fait entendre chaque fois qu'une touche est frappée peut être inhibé lui-aussi, de deux façons: la première, classique, consiste à baisser le son de votre récepteur T.V., la seconde à forcer à 1 le registre BUZZ.

B retourne le code ASCII du caractère. Si aucune touche n'a été enfoncée, ou si l'une des deux touches o ou CNT a été enfoncée seule, ou si plusieurs touches parmi les touches o ou CNT ont été enfoncées simultanément, B retourne la valeur zéro.

La touche o sélectionne le caractère se trouvant en haut de la touche. La touche CNT force à 0 le bit 6 du code ASCII du caractère de la touche; dans ce cas les minuscules sont forcées en majuscules.

L'accès aux minuscules accentuées nécessite en général trois frappes consécutives:

- 1° frappe: touche accent.

- 2° frappe: touche o en même temps que la touche représentant l'accent.

- 3° frappe: lettre minuscule.

Cependant, l'accès à certaines minuscules accentuées couramment utilisées en Francais peut être fait en deux touches:

* ACC/6: é

* ACC/7: è

* ACC/8: ù

* ACC/9: ç

* ACC/0: à

Mais dans tous les cas, vous devrez faire trois appels consécutifs à cette routine:

- le 1° appel retourne dans B le code de la touche ACC, soit 16H.

- le 2° appel retourne dans B le code de l'accent ou de la cédille.

- le 3° appel retourne dans B le code de la minuscule.

Remarque: bien que ACC suivi de C produise un à, et ACC suivi de o/O produise un accent grave, la frappe de o/O non précédée de la frappe de ACC retournera O dans B (pas de caractère connu), car la quote inverse n'est pas reconnue par le TO7-70.

Le code ASCII du caractère est lu dans une table dont l'adresse se trouve dans le registre PTCLAV (60CDH-60CEH). Si vous voulez redéfinir votre clavier, il suffira de mettre dans ce registre l'adresse de la table contenant vos propres codes ASCII, mais les séquences "accent" peuvent donner des résultats imprévisibles ...

7.0 PRIMITIVES GRAPHIQUES

7.1 Allumage ou extinction d'un point

* adresse du point d'entrée: PLOT$ (E80FH)

* paramètres d'entrée:

- registres 6809 X et Y

- registres FORME (6038H), CHDRAW (6041H), COLOUR (603BH) et STATUS (6019H)

* paramètres de retour:

- registres PLOTX (603DH-603EH) et PLOTY (603FH-6040H)

7.1.1 Mode graphique

Cette routine met en couleur le point de coordonnées passées par X et Y, où X C <0,319> et Y C <0,199>. La couleur du point est déterminée par le contenu du registre 6038H, qui doit être compris entre -8 et +15 pour le **TO7-70**. Si le code de la couleur est négatif, le point sera écrit en "fond", c'est-à-dire que le bit correspondant dans l'octet en mémoire "caractère" sera mis à zéro. Si le code est positif, le point sera écrit en "forme", c'est-à-dire que le bit correspondant dans l'octet en mémoire "caractère" sera mis à 1. Sur le **TO7-70**, si le bit 4 du registre STATUS est à 1, seul le bit en mémoire caractère sera modifié, la couleur ne sera pas changée.

Voici les codes des couleurs "forme" ou "fond":

COULEUR	CODE "FORME"	CODE "FOND"
NOIR	0	-1
ROUGE	1	-2
VERT	2	-3
JAUNE	3	-4
BLEU	4	-5
MAGENTA	5	-6
CYAN	6	-7
BLANC	7	-8
GRIS	8	
ROSE	9	
VERT CLAIR	10	
SABLE	11	
BLEU CLAIR	12	
PARME	13	
BLEU CIEL	14	
ORANGE	15	

Un code de couleur "fond" se déduit donc d'un code de couleur "forme" en ajoutant 1 et en prenant l'opposé.

Le registre CHDRAW (6041H) doit être mis à zéro, sinon l'on est dans le mode caractère, explicité ci-dessous.

7.1.2 Mode caractère

Si le contenu du registre CHDRAW est non nul, il est supposé contenir le code ASCII d'un "point caractère" à afficher à l'endroit défini par les registres X et Y. Dans ce cas, on doit avoir X C <1,40> et Y C <0,24>.

La couleur du "point caractère" est fournie non plus par le registre FORME (6038H), mais par le registre COLOUR (603BH), qui contient les couleurs courantes de "fond" et de "forme".

On peut accèder directement à l'écriture du caractère du registre CHDRAW (6041H) en faisant un appel à l'adresse CHPL$ (E833H).

Dans les deux modes, les registres X et Y seront recopiés respectivement dans les registres PLOTX (603DH-603EH) et PLOTY (603FH-6040H) qui gardent l'abscisse et l'ordonnée du dernier point allumé.

7.2 Tracé d'un segment de droite

* Adresse du point d'entrée: DRAW$ (E80CH)

* Paramètres d'entrée:

- registres 6809 X et Y

- registres PLOTX (603DH-603EH), PLOTY (603FH-6040H), CHDRAW (6041H), FORME (6038H), COLOUR (603BH), et STATUS (6019H)

* paramètres de retour:

- registres PLOTX (603DH-603EH), PLOTY (603FH-6040H)

7.2.1 Mode graphique

Cette routine trace un segment de droite entre le point défini par son abscisse en PLOTX et son ordonnée en PLOTY, (dernier point allumé si vous n'avez pas modifié ces registres depuis), et le point dont les coordonnées sont passées par X et Y, avec X C <0,319> et Y C <0,199>. La couleur du segment est définie par le contenu du registre FORME, suivant les conventions explicitées ci-dessus.

Si le bit 4 du registre STATUS (6019H) est mis à 1, la couleur ne sera pas mise à jour. Le contenu du registre CHDRAW doit être nul, sinon le segment tracé est un "segment caractère".

7.2.2 Mode caractère

Si le contenu du registre CHDRAW n'est pas nul, il est interprèté comme le code ASCII du caractère servant au tracé. Dans ce cas, la couleur est la couleur courante donnée par le registre COLOUR, et les coordonnées X et Y doivent être comprises respectivement dans les intervalles <1,40> et <0,24>.

Dans les deux modes, les registres X et Y seront recopiés respectivement dans les registres PLOTX et PLOTY, qui gardent l'abscisse et l'ordonnée du dernier point allumé, ce qui permet de tracer un contour polygonal sans avoir à préciser à chaque fois le point de départ.

7.3 Segments horizontaux

Dans le cas où le segment à tracer est horizontal, un algorithme de tracé rapide est mis en oeuvre, ce qui peut être utile pour remplir des contours par une série de segments horizontaux.

7.4 Lecture de la couleur d'un point

* Adresse du point d'entrée: GETP$ (E821H)

* paramètres d'entrée:

- registres 6809 X et Y

* paramètre de retour:

- registre 6809 B

Cette routine retourne la couleur d'un point dans l'accumulateur B, de -16 à -1 si le point est en "fond", et de 0 à +15 si le point est en "forme". Les coordonnées du point sont passées par X C <0,319> et par Y C <0,199>. La couleur est codée comme décrit au § 7.1.1

8.0 LECTURE DE L'ECRAN

* Adresse du point d'entrée: GETS$ (E824H)

* paramètres d'entrée:

- registres 6809 A et X

* paramètres de retour:

- registre 6809 B

Cette routine retourne dans l'accumulateur B le code ASCII du caractère dont les coordonnées sont passées par X C <1,40> et A C <0,24>.

S'il n'y a pas de caractère connu en (X,A) alors B retourne la valeur zéro. Sinon, deux cas peuvent se produire:

8.1 Caractère normal

Si le caractère appartient à l'alphabet G0, alors B retourne le code ASCII du caractère reconnu.

8.2 Minuscule accentuée ou ç

Dans ce cas, trois appels à la routine sont nécessaires:

- 1° appel: une minuscule accentuée est détectée; B retourne le code ACC (accent), soit 16H.

- 2° appel: B retourne le code de l'accent.

3° appel: B retourne le code ASCII de la minuscule.

9.0 GENERATION DE MUSIQUE

* Adresse du point d'entrée: NOTE$ (E81EH)

* paramètres d'entrée:

- registre 6809 B

- registres OCTAVE (6036H-6037H), DUREE (6033H-6034H), TEMPO (6031H-6032H), et TIMBRE (6035H)

La note à jouer est passée par l'accumulateur B. Il y a 13 notes de base, de DO à UT, plus le silence. Voici la table des valeurs correspondant aux notes:

NOTE:	CODE:
SILENCE	30H
DO	31H
DO≠	32H
RE	33H
RE≠	34H
MI	35H
FA	36H
FA≠	37H
SOL	38H
SOL≠	39H
LA	3AH
LA≠	3BH
SI	3CH
UT	3DH

Il faut ensuite préciser les paramètres suivants: tempo, octave, timbre et durée.

* l'octave: il y a 5 octaves possibles, de l'octave 1 qui est la plus grave, à l'octave 5. L'octave 4 correspond à l'octave du LA 440.

Voici la liste des valeurs à mettre dans le registre OCTAVE :

OCTAVE:	VALEUR:
1	16
2	08
3	04
4	02
5	01

* la durée: il s'agit de la durée relative de chaque note, pouvant aller de la ronde à la triple croche. La valeur de la durée est à charger dans le registre DUREE. La valeur pour la ronde est 96, et l'on obtient les valeurs relatives en divisant par des puissances de 2, ou de 3 pour des triolets ou les notes pointées. Voici la liste des valeurs:

NOTES:	VALEURS:
RONDE	96
BLANCHE pointée	72
BLANCHE	48
NOIRE pointée	36
NOIRE	24
CROCHE pointée	18
CROCHE	12
DOUBLE croche pointée	09
DOUBLE croche	06
TRIPLE croche pointée	05
TRIPLE croche	03

DANS UN TRIOLET:	VALEURS:	
NOIRE	16	(car 3x16 = 48)
CROCHE	08	(car 3x8 = 24)
DOUBLE CROCHE	04	(car 3x4 = 12)
TRIPLE CROCHE	02	(car 3x2 = 06)

* le tempo: c'est le mouvement auquel doit être joué le morceau. La durée réelle d'une note est égale au tempo x la durée. La valeur du tempo, de 1 à 255, doit être chargée dans le registre TEMPO.

* le timbre: cette valeur (de 0 à 5) doit être chargée dans le registre TIMBRE. Le rapport cyclique est modifié en conséquence, ce qui donne à la note une attaque différente. Pour une note continue, il faut mettre la valeur 0 dans le registre.

10.0 LECTURE DES MANETTES DE JEU

* Adresse du point d'entrée: JOYS$ (E827H)

* paramètre d'entrée:

- registre 6809 A

* paramètres de retour:

- registres 6809 B et CC

Le numéro de la manette de jeu dont on veut connaître l'état (0 ou 1) est passé par l'accumulateur A.

B retourne une valeur de 0 à 8 donnant l'état de la manette de jeu, suivant la convention que voici:

0 ==> Centre

1 ==> Nord

2 ==> Nord Est

3 ==> Est

4 ==> Sud Est

5 ==> Sud

6 ==> Sud Ouest

7 ==> Ouest

8 ==> Nord Ouest

Le bit de retenue du registre CC est mis à 1 si le bouton a été enfoncé, sinon il est mis à zéro.

11.0 CRAYON OPTIQUE

11.1 Test du bouton du crayon optique

* Adresse du point d'entrée: LPIN$ (E81BH)

* paramètre de retour:

- registre 6809 CC

Cette routine teste l'état du bouton du crayon optique. Si celui-ci a été enfoncé, le bit de retenue est forcé à 1, sinon il est forcé à 0.

11.2 Lecture du crayon optique

* Adresse du point d'entrée: GETL$ (E818H)

* paramètres de retour:

- registres 6809 X,Y et CC

Cette routine lit les coordonnées du point visé par le crayon optique, et retourne l'abscisse dans X <0,319> et l'ordonnée dans Y <0,199>. Si la mesure est correcte, le bit de retenue du registre CC est forcé à zéro. En cas de mauvaise lecture (luminosité trop faible, crayon trop éloigné de l'écran), ce bit est forcé à 1.

12.0 GESTION DE L'INTERFACE DE COMMUNICATION

* Adresse du point d'entrée: RSCO$ (E812H)

* paramètres d'entrée:

- registre 6809 B

- registres RS.OPC (602BH) ,BAUDS (6044H-6045H), NOMBRE (6046H) et GRCODE (6047H)

* paramètres de retour:

- registre 6809 CC

- registres RS.STA (602CH)

Cette routine gère l'interface de communication. Le contenu du registre RS.OPC sélectionne l'une des opérations suivantes:

CONTENU DE RS.OPC	OPERATION DEMANDEE
% 0 0 0 0 0 0 0 1	Ouverture en lecture écriture (RS232)
% 0 0 0 0 0 0 1 0	Lecture d'un caractère (RS232)
% 0 0 0 0 0 1 0 0	Ouverture en écriture seule (RS232)
% 0 0 0 0 1 0 0 0	Ecriture d'un caractère
% 0 0 0 1 0 0 0 0	Fermeture
% 0 0 1 0 0 0 0 0	Copie graphique d'écran
% 0 1 0 0 0 0 0 0	Ouverture en écriture en parallèle

En cas d'écriture, B doit contenir l'octet à envoyer.

En cas de copie graphique de l'écran, le registre GRCODE doit contenir le code de mise en mode graphique spécifique de l'imprimante. Ce registre contient la valeur 7 par défaut.

Le registre RS.STA retourne le code de l'opération réalisée, et en cas d'erreur, retourne un des codes suivants :

CONTENU DE RS.STA	ETAT DE LA COMMUNICATION
% 0 0 0 0 0 0 0 1	Ouvert en lecture écriture (RS232)
% 0 0 0 0 0 1 0 0	Ouvert en écriture seule (RS232)
% 0 0 0 1 0 0 0 0	Fermé
% 0 1 0 0 0 0 0 0	Ouvert en écriture en parallèle
% 1 0 0 0 0 0 0 0	Périphérique non prêt

Le bit de retenue du registre CC est forcé à zéro si tout s'est passé normalement, sinon il est forcé à 1.

Pour les transmissions série, le registre NOMBRE doit contenir la valeur 80H si 8 bits de données doivent être transmis pour un octet, ou 40H si seulement 7 bits sont à transmettre.

La vitesse de transmission est sélectable de 110 à 4800 bauds, par un paramètre mis dans le registre BAUDS. Ce paramètre est pris dans la table BDTAB située à l'adresse E836H. Le premier paramètre (sur 2 octets) représente la vitesse pour 110 bauds, les suivants pour 300, 600, 1200, 2400 et 4800 bauds.

13.0 GESTION DU LECTEUR-ENREGISTREUR DE CASSETTES

* adresse du point d'entrée: K7CO$ (E815H)

* paramètres d'entrée;

- registres 6809 B

- registres K7.OPC (6029H)

* paramètres de retour:

- registre 6809 CC

- registres K7.STA (602AH)

Cette routine permet de lire ou d'écrire des données sur la cassette selon l'état du registre K7.OPC.

CONTENU DE K7.OPC	OPERATION DEMANDEE
% 0 0 0 0 0 0 0 1	Ouverture en lecture
% 0 0 0 0 0 0 1 0	Lecture d'un caractère
% 0 0 0 0 0 1 0 0	Ouverture en écriture
% 0 0 0 0 1 0 0 0	Ecriture d'un caractère
% 0 0 0 1 0 0 0 0	Fermeture

Dans le cas d'une écriture, B doit contenir l'octet à envoyer.

Le registre K7.STA retourne le code de l'operation réalisée, et en cas d'erreur, positionne le bit de retenue du CC à 1 et retourne un des codes suivants :

CONTENU DE K7.STA	ETAT DE LA CASSETTE
% 0 0 0 0 0 0 0 1	Ouvert en lecture
% 0 0 0 0 0 1 0 0	Ouvert en écriture
% 0 0 0 1 0 0 0 0	Fermé
% 1 0 0 0 0 0 0 0	Périphérique non prêt

Si le L.E.P. n'est pas raccordé à votre TO7-70, le bit de retenue du CC sera forcé à 1.

14.0 CONTROLEUR DE DISQUETTES

Vous pouvez gérer les entrées/sorties disque à deux niveaux: au niveau physique en utilisant le point d'entrée du moniteur, ou à un niveau logique en manipulant des fichiers au format Basic Microsoft(R).

Rappelons que les disquettes sont divisées en 40 pistes de 16 secteurs chacune. Un secteur contient lui-même 128 octets en simple densité, ou 256 en double densité.

De plus, le contrôleur double densité peut fonctionner dans les 2 modes : simple ou double densité. Au démarrage, sa densité de travail doit être sélectionnée après l'initialisation du contrôleur.

14.1 Entrées/Sorties disque physiques

* Adresse du point d'entrée: DKCO$ (OE82AH)

* paramètres d'entrée:

- registres DK.OPC (6048H), DK.DRV (6049H), DK.SEC (604CH), DK.TRK (604AH-604BH), DK.BUF (604FH-6050H)

* paramètres de retour:

- registre 6809 CC

- registre DK.STA (604EH)

La présence du contrôleur de disquette est testée par la mise à FFH du registre DK.FLG (6080H). Si ce registre est nul, le contrôleur est absent.

En cas d'erreur pendant une opération le bit de retenue du CC est forcé à 1, sinon il est forcé a 0.

Le registre DK.OPC contient le code de l'opération à réaliser:

- Code 01: demande l'initialisation du contrôleur. Dans ce cas, si l'initialisation a pu avoir lieu sans erreurs, le bit de retenue du registre code condition (CC) est mis à 0, et le type de contrôleur est retourné dans le registre DK.STA: "C" pour la simple densité, et "D" pour la double densité. Sinon, le bit de retenue est mis à 1, et le code d'erreur 40H est mis dans le registre DK.STA. (voir ci-après les codes d'erreur).

- Code 02: lecture d'un secteur. Les codes d'erreur possibles sont 02, 04, 08, 10H, 80H. Si une erreur a eu lieu, son code est retourné dans le registre DK.STA.

- Code 04: sur lecteur double densité, passage du contrôleur en simple densité, pas d'erreur possible. Erreur sur un lecteur simple densité.

- Code 08: écriture d'un secteur. Les codes d'erreur possibles sont 01, 02, 04, 08, 10H, 20H, 80H. Si une erreur a eu lieu, son code est retourné dans le registre DK.STA.

- code 10H: sur lecteur double densité, passage du contrôleur en double densité, pas d'erreur possible. Erreur sur un lecteur simple densité.

- Code 20H: recherche de la piste zéro. Les codes d'erreur possibles sont 10H et 80H.

- Code 40H: recherche la piste dont le numéro est donné par le registre DK.TRK. Les codes d'erreur possibles sont 10H et 80H.

- Code 80H: option de vérificationen écriture. Il faut faire un "OU" logique entre ce code et le code de l'opération que l'on veut vérifier. Les codes d'erreur retournés sont les mêmes que ceux de l'operation specifiée augmentés du code 20H.

Paramètres à passer suivant l'opération demandée:

* registre DK.DRV (6049H): ce registre doit contenir le numéro du lecteur concerné, soit une valeur entre 0 et 3.

* registre DK.TRK (604AH-604BH): ce registre doit contenir le numéro de piste, les pistes étant numerotées de 0 à 39.

* registre DK.SEC (604CH): ce registre doit contenir le numéro de secteur où l'on veut lire ou écrire. Ce numéro doit être compris entre 1 et 16.

* registre DK.BUF (604FH-6050H): ce registre doit contenir l'adresse de début d'une zone tampon en RAM de 128 octets en simple densité, ou de 256 octets en double densité, soit pour y lire les données à écrire sur disque, soit pour y écrire les données lues sur le disque.

* registre DK.STA (604EH): ce registre contient le code d'erreur, ou le type de contrôleur après une initialisation correcte.

Voici les différentes erreurs possibles:

- code 01: disquette protégée; cette erreur ne peut apparaître qu'après une demande d'écriture.

- code 02: erreur de piste; l'identificateur de piste est correct, mais ne correspond pas à la piste demandée.

- code 04: erreur de secteur; l'identificateur de secteur est incorrect (secteur ne pouvant être lu ou erreur sur le checksum), cependant la piste peut être correcte.

- code 08: erreur sur les données; l'identificateur de secteur est correct, mais les données ne peuvent être lues, ou le checksum est incorrect.

- code 10H: lecteur non prêt; le moteur n'est pas en route, ou le lecteur specifié est inexistant.

- code 20H: erreur sur vérification; la zone tampon en mémoire et la zone correspondante écrite sur la disquette ne sont pas identiques.

- code 40H: contrôleur non prêt.

- code 80H: disquette non formattée. L'identificateur de piste ne peut être lu.

14.2 Le format Basic Microsoft(R)

Pour assurer la compatibilité entre les diverses applications, les fichiers créés par le TO7-70 suivent le standard Basic Microsoft(R).

La piste 20 est une zone réservée destinée à rendre compte de l'état de la disquette. Elle est organisée comme suit:

* secteur 1: réservé

* secteur 2: table d'allocation des fichiers ou "FAT"

* secteurs 3 à 16: catalogue

14.2.1 La table d'allocation des fichiers

Les fichiers sont organisés en blocs de 1K octets en simple densité, ou 2K octets en double densité. On a donc dans tous les cas 2 blocs par piste. Les blocs sont numerotés à partir de 0. Chaque octet de la table d'allocation des fichiers, à partir de l'octet 1, représente un bloc physique.

Organisation de la "FAT":

* Octet 0: 0.

* Octet 1: bloc 0, piste 0, secteurs 1 à 8.

* Octet 2: bloc 1, piste 0, secteurs 9 à 16.

* Octet 3: bloc 2, piste 1, secteurs 1 à 8.

* Octet 4: bloc 3, piste 1, secteurs 9 à 16.

* .....

* Octet 2j-1: bloc 2j-2, piste j-1, secteurs 1 à 8.

* Octet 2j: bloc 2j-1, piste j-1, secteurs 9 à 16.

* .....

* Octet 80: bloc 79, piste 39, secteurs 9 à 16.

Un octet de la "FAT" représentant un bloc physique peut avoir comme valeurs:

* FFH, qui signifie bloc non alloué.

* FEH, qui signifie bloc réservé.

* Tout nombre de 0 à BFH, signifie bloc alloué. Dans ce cas, le nombre représente le numéro du bloc logique suivant du même fichier.

* Tout nombre de C1H à C8H, signifie dernier bloc d'un fichier. Les 4 bits de poids faible indiquent le nombre de secteurs utilisés dans ce dernier bloc.

14.2.2 Le catalogue

Le catalogue donne la liste des fichiers, et occupe 14 secteurs. Chaque fichier est répertorié sur 32 octets. Il y a donc 4 fichiers répertoriés par secteur en simple densité, et 8 fichiers par secteur en double densité. Au total, le catalogue peut donc répertorier 56 fichiers en simple densité, et 112 en double densité.

Chaque fichier est répertorié de la façon suivante:

Octets 00 à 07: Nom du fichier, cadré à gauche, complété par des blancs.

Octets 08 à 0AH: Suffixe du fichier (.BAS, .BIN, etc...), cadré à gauche, complété par des blancs.

Octet 0BH: Type de fichier: 0 pour un programme Basic ASCII ou binaire, 1 pour des données Basic en ASCII, 2 pour un programme en langage machine (binaire), 3 pour un fichier assembleur édité en ASCII.

Octet 0CH: Sémaphore: FFH pour de l'ASCII, 00 pour du binaire.

Octet 0DH: Numéro du premier bloc logique du fichier.

Octets 0EH-0FH: Nombre d'octets utilisés dans le dernier secteur du fichier.

Octets 10H-1FH: Réservés.

Le premier octet de chaque entrée dans le catalogue indique son état:

* 00: entrée non allouée, pas de fichier répertorié pour cette entrée.

* 20H-7FH: code ASCII du premier caractère du nom de fichier, donc entrée allouée.

* FFH: fin logique du catalogue.

Lors de la création du catalogue, ces octets sont mis à FFH. Chaque fois qu'un fichier est créé, la fin logique du catalogue est déplacée dans le premier octet de l'entrée suivante, jusqu'à ce que le catalogue soit plein. Lorsqu'un fichier est détruit, le premier octet de son entrée est mis à zéro (entrée non allouée). Dans ce cas, tout fichier nouvellement créé se verra attribuer en priorité cette entrée.

15.0 INFORMATIONS COMPLEMENTAIRES

15.1 Organisation de la mémoire

ADRESSES (HEXADECIMAL)	
C000-3FFF	Cartouche ROM
4000-5FFF	2 x 8 K-Octets de mémoire écran
6000-60FF	Registres moniteurs
6100-DFFF	Mémoire utilisateur dont 16 K commutables
A000-DFFF	Banque de 16K utilisateur commutable
A000-DFFF	Extension 4 banques de 16 K utilisateur
E000-E7BF	1.9 K pour le disque
E7C0-E7C7	PIA 6846 système
E7C8-E7CB	PIA 6821 système
E7CC-A7CF	PIA 6821 extension jeux
E7D0-E7DF	Contrôleur de floppy
E7E0-E7E3	PIA 6821 interface de communication
E7E4-E7E7	Compteurs crayon optique
E7E8-E7FF	Extensions
E800-FFFF	6 K-Octets moniteur

La partie de mémoire située entre A000H et DFFFH est commutable avec d'autres partie de RAM. Si vous avez le TO7-70 de base, vous ne disposez que de 2 banques commutables plus 16 K stables (de 6000H à 9FFFH) soit 48 K au total, dont 32 K accessibles en même temps. Si vous disposez de l'extension mémoire 64 K, vous disposerez de 6 banques de 16 K commutables. La commutation des banques se fait par le sous-programme suivant :

Entrée : registre 6809 A = Numéro de banque 0 à 5

```
COMMUT  EQU     *
        PSHS    D,X,U
        LDU     #0E7C0H
        LDB     11,U
        ANDB    #0FBH
        STB     11,U
        LDX     #TAB
        LDA     A,X
        STA     9,U
        ORB     #04H
        STB     11,U
        PULS    D,X,U,PC
TAB     EQU     *
        FCB     0FH,17H,0E7H,67H,0A7H,27H
```

15.2 Points d'entrée standard du moniteur

PUTC$ (E803H) : affichage d'un caractère.

GETC$ (E806H) : lecture du clavier.

KTST$ (E809H) : lecture rapide du clavier.

DRAW$ (E80CH) : tracé d'un segment de droite.

PLOT$ (E80FH) : allumage ou extinction d'un point.

RSCO$ (E812H) : gestion de l'interface de communication.

K7CO$ (E815H) : lecture/écriture sur la cassette.

GETL$ (E818H) : lecture du crayon optique.

LPIN$ (E81BH) : lecture du bouton du crayon optique.

NOTE$ (E81EH) : génération de musique.

GETP$ (E821H) : lecture de la couleur d'un point.

GETS$ (E824H) : lecture de l'écran.

JOYS$ (E827H) : lecture des manettes de jeu.

DKCO$ (E82AH) : contrôleur de disque.

MENU$ (E82DH) : retour au menu principal

KBIN$ (E830H) : sortie programme d'interruption

CHPL$ (E833H) : écriture d'un point "caractère".

15.3 Registres du moniteur

Les adresses qui suivent sont données en hexadécimal:

* 6000-6018 (TERMIN) : Table des terminateurs de lignes Basic.
* 6019 (STATUS) : Différents sémaphores:

b7: sémigraphique
b6: scroll rapide
b5: interruption utilisateur
b4: graphiques sans écriture de couleur
b3: sémaphore de lecture clavier
b2: curseur visible/invisible
b1: réservé
b0: touche clavier déjà lue

* 601A-601B (TABPT) : Pointeur dans la table des terminateurs de lignes.
* 601B (RANG) : Ligne logique courante.
* 601C-601D (TOPTAB) : Pointeur sur le sommet logique de la table des terminateurs de lignes.
* 601D (TOPRAN) : Première ligne logique de la fenêtre.
* 601E-601F (BOTTAB) : Pointeur sur la fin logique de la table des terminateurs de ligne.
* 601F (BOTRAN) : Dernière ligne logique de la fenêtre.
* 6020 (COLN) : Colonne logique courante.
* 6021-6022 (IRQPT) : Pointeur sur la routine moniteur de traitement des interruptions IRQ.
* 6023-6024 (FIRQPT) : Pointeur sur la routine de traitement des interruptions rapides FIRQ.
* 6025-6026 (NMIPT) : Pointeur sur l'interruption NMI.
* 6027-6028 (TIMEPT) : Pointeur sur la routine utilisateur de traitement des interruptions utilisateur.
* 6029 (K7.OPC) : Code opération du lecture-enregistreur de programmes (LEP).
* 602A (K7.STA) : Code état du LEP.
* 602B (RS.OPC) : Mot de commande pour la gestion de la communication.
* 602C (RS.STA) : Etat courant de la liaison communication.
* 602D-602E (USERAF) : Pointeur sur le générateur de caractères utilisateur.
* 602F-6030 (SWI1) : Pointeur sur SWI.
* 6031-6032 (TEMPO) : Tempo général pour la génération de musique.
* 6033-6034 (DUREE) : Durée de la note (de 1 à 96).
* 6035 (TIMBRE) : Attaque de la note.
* 6036-6037 (OCTAVE) : Octave (1, 2, 4, 8 ou 16).
* 6038 (FORME) : Contient le code de la couleur de -8 à +15 pour la mise en couleur d'un point ou le tracé d'un segment de droite.
* 6039 (ATRANG) : Sémaphores pour la gestion d'écran.

b7: sémaphore de scroll
b6: Réservé
b5: Réservé
b4: Réservé
b3: Réservé
b2: Réservé
b1: largeur simple ou double
b0: hauteur simple ou double

* 603A (ATRSCR) : Sémaphores pour la gestion plein écran.

b7: sémaphore de fond plein écran
b6: sémaphore de forme plein écran
b5: Réservé
b4: Réservé
b3: Réservé
b2: Réservé
b1: largeur simple ou double
b0: hauteur simple ou double

* 603B (COLOUR) : Couleur courante ; les 3 bits de poids faible donnent la couleur du fond, les 3 bits suivants la couleur de la forme, suivant le codage video BVR. Les 2 bits de poids fort représentent les couleurs pastels.
* 603C (TELETL) : Si ce registre contient la valeur FFH, on est en mode "page" (pas de scroll).
* 603D-603E (PLOTX) : Abscisse du dernier point allumé ou éteint.
* 603F-6040 (PLOTY) : Ordonnee du dernier point allumé ou éteint.
* 6041 (CHDRAW) : Code ASCII du caractère pour un tracé de point ou de droite en mode "caractères". Si ce registre contient la valeur 0, le tracé est fait en mode "points".
* 6042 (CURSFL) : Sémaphore de mouvement curseur, qui, s'il contient la valeur 255, indique qu'il ne faut pas lier logiquement la ligne à la suivante.
* 6043 (COPCHR) : Sémaphore qui, s'il contient la valeur 255, indique que le déplacement à droite ou à gauche recopie le caractère courant.
* 6044-6045 (BAUDS) : Paramètre de vitesse de la liaison série.
* 6046 (NOMBRE) : Paramètre du nombre de bits de la liaison série (8=80H, 7=40H).
* 6047 (GRCODE) : Mot de code pour la mise en mode graphique de l'imprimante.
* 6048 (DK.OPC) : Mot de commande pour le contrôleur de disques.
* 6049 (DK.DRV) : Numéro de lecteur de disquettes.
* 604A-604B (DK.TRK) : Numéro de piste.
* 604C (DK.SEC) : Numéro de secteur.
* 6D4D (DK.NUM) : Entrelacement de secteurs.
* 604E (DK.STA) : Etat du contrôleur de disquettes.
* 604F-6050 (DK.BUF) : Pointeur sur la zone-tampon reservée aux entrées/sorties disque.
* 6051-6052 (TRACK0) : Position de la tête du lecteur 0.
* 6053-6054 (TRACK1) : Position de la tête du lecteur 1.
* 6055-6056 (TRACK2) : Position de la tête du lecteur 2.
* 6057-6058 (TRACK3) : Position de la tête du lecteur 3.
* 6059 (SEQUCE) : Code indiquant dsns quelle séquence de gestion d'écran on se trouve.
* 605A-605B (SCRPT) : Pointeur courant dans l'écran.
* 605C (SAVCOL) : Sauvegarde de la couleur courante.
* 605D (ASCII) : Code du dernier caractère affiché .
* 605E (KEY) : Dernière touche frappée.
* 605F (CMPTKB) : Compteur de répétitions clavier.
* 6060-6061 (STADR) : Adresse du premier octet de la fenêtre.
* 6062-6063 (ENDDR) : Adresse + 1 du dernier octet de la fenêtre.
* 6064 (TCRSAV) : Sauvegarde de l'état courant du timer.
* 6065-6066 (TCTSAV) : Sauvegarde du compte courant du timer.
* 6067 : Réservé.
* 6067 (LATCLV) : Latence clavier.

```
* 6068-6069 (SAVATR) : Sauvegarde des attributs courants d'écran.
* 606A (US1) :         Sémaphore pour les séquences "unit separator"
* 606B (COMPT) :       Compteur de caractère répétés.
* 606C-606D (TEMP) :   Registre temporaire pour le transfert de
                       données.
* 606E-606F (SAVEST) : Sauvegarde du pointeur de pile.
* 6070 (ACCENT) :      Sémaphore pour les séquences accent.
* 6071 (SS2GET) :      Sémaphore pour l'impression ou la lecture
                       d'une minuscule accentuée.
* 6072 (SS3GET) :      Sémaphore pour l'impression ou la lecture
                       d'une minuscule accentuée.
* 6073 (BUZZ) :        Sémaphore d'extinction du buzzer.
* 6074 :               Réservé.
* 6075 (EFCMPT) :      Compteur d'effacements du curseur.
* 6076-6077 (BLOCZ) :  Deux octets toujours à la valeur zéro pour les
                       initialisations.
* 6078 (SCROLS) :      Sémaphore de scroll doux.
* 6079-607E (TABCHX) : Tables de pointeurs du menu.
* 607F (RUNFLG) :      Sémaphore indiquant que l'option 2 a été
                       choisie.
* 6080 (DKFLG) :       Sémaphore de présence du contrôleur disques.
* 6081-60CC :          Pile système.
* 60CD-60CE (PTCLAV) : Pointeur sur la table de décodage du clavier.
* 60CF-60D0 (PTGENE) : Pointeur sur le générateur de caractères
                       standard.
* 60D1 (APPLIC) :      Checksum de l'application en cours.
* 60D2 (DECALG) :      Ajustement pour le crayon optique.
* 60D3-60EA (LPBUFF) : Zone-tampon pour la lecture du crayon optique.
* 60E5-60FD :          Réservés.
* 60FE-60FF (TSTRST) : Sémaphore de démarrage à chaud ou à froid.
```

15.4 Adresses d'Entrées/Sorties

Les adresses qui suivent sont données en hexadécimal.

15.4.1 PIA système

*** PIA SYSTEME 6846 ***

* E7C0 (CSR): Registre d'état

* E7C1 (CRC): Registre de contrôle :

CC2 : sortie son
CT0 : écriture cassette

* E7C2 (DDRC): Registre de direction

* E7C3 (PRC): Registre de données:

bit0 (sortie): commutation mémoire écran caractères et mémoire écran couleur.
bit1 (entrée): interrupteur crayon optique.
bit2 (sortie): couleur du tour, pastel.
bit3 (sortie): led clavier.
bit4 (sortie): couleur du tour, rouge.
bit5 (sortie): couleur du tour, vert.
bit6 (sortie): couleur du tour, bleu.
bit7 (entrée): lecture cassette.

* E7C5 (TCR): Registre contrôle timer

* E7C6-E7C7 (TMSB-TLSB): Valeur timer

*** PIA SYSTEME 6821 ***

* E7C8 (PRA): Registre de données, port A.

bit0-7 (entrée): lecture matrice clavier

* E7C9 (PRB): Registre de données, port B:

bit 0-2 (sortie): multiplexage clavier
bit 3-7 (sortie): sélection banques mémoire

* E7CA (CRA): Registre de contrôle, port A:
CA1 (entrée): présence carte incrustation
CA2 (sortie): moteur du L.E.P.

* E7CB (CRB): Registre de contrôle, port B:
CB1 (entrée): interruption light-pen
CB2 (sortie): commande d'incrustation.

15.4.2 PIA jeux

*** PIA JEUX 6821 ***

* E7CC (PRA1): Registre de données, port A:

bits 0-7 (entrée): lecture des manettes de jeu.

* E7CD (PRB1): Registre de données, port B:

bits 0-5 (entrée): convertisseur digital/analogique.
bit6 (entrée): action manette de jeu 0.
bit7 (entrée): action manette de jeu 1.

* E7CE (CRA1): Registre de contrôle, port A:
CA1 (entrée): action manette de jeu 0.

* E7CF (CRB1): Registre de contrôle, port B:
CB1 (entrée): action manette de jeu 1.

15.4.3 Interface de communication

*** INTERFACE DE COMMUNICATION 6821 ***

* E7E0 (PRA2): Registre de données, port A:

bit0 (sortie): receive data
bit1 (sortie): clear to send
bit5 (entrée): request to send
bit6 (entrée): data terminal ready
bit7 (entrée): transmit data

* E7E1 (PRB2): Registre de données, port B:

bits 0-7 (sortie): données en parallèlle.

* E7E2 (CRA2): Registre de contrôle, port A:
CA1 (entrée): request to send (demande d'émission).

* E7E3 (CRB2): Registre de contrôle, port B:
CB1 (entrée): acknowledge.
CB2 (sortie): strobe.

Achevé d'imprimer le 14 mai 1985 sur les presses de l'Imprimerie
Carlo Descamps à Condé-sur-l'Escaut 59163 France

Dépôt légal : mai 1985 — N° d'imprimeur : 3804
Imprimé en France